MEMORY ACCESS REORDERING

Deepika Krishnaswamy
Harini Raghunathan
Iswari Natarajan
Outline

• Access Reordering – What? Why? …
• Our Approach
• Performance Modelling
  • RDRAM
  • SDRAM
• Applications – Where is it important?
• Conclusion
How do you get data?

Read Request
Physical address

Smart Memory Controller
32 bit physical address

Rank id = ?
Bank id = ?
Row id = ?
Col id = ?
Rank, Bank ...

Bank id = 1

Rank id = 1

Column id = 0x187

Row id = 0x0B1D
Banks are independent

Multiple banks reduce access conflicts

Bank Conflict

Access 1: Rank id 0, Bank id 0, Row id 235
Access 2: Rank id 0, Bank id 0, Row id 354
Access 3: Rank id 0, Bank id 0, Row id 235
Access Reordering reduces bank conflict

Access 1: Rank id 0, Bank id 0, Row id 235
Access 3: Rank id 0, Bank id 0, Row id 235
Access 2: Rank id 0, Bank id 0, Row id 354

Strict Ordering

Memory Access Re-ordered

ACT – Activate Page  READ – Read Data  PRE - Precharge
Method of Implementation

- Simulator used – Simplescalar
- Simplescalar models a virtual computer system which includes the CPU and DRAM memory system.
Priority Schemes

Different schemes to pick next memory access request from the BIU

- FCFS (First Come First Serve)
- RIFF (Read or Ifetch First)
- REORDER
Reorder Scheme

Pick next oldest request from BIU

Bank Conflict?

Yes

Trace the entire BIU to look for a bank hit

No

Bank Hit?

Yes

No

Service Request

No
Performance Modelling

- **RDRAM (Reorder vs. FCFS and RIFF)**
  - No. of Bank Conflicts
  - No. of column accesses per row access
  - Latency

- **SDRAM (Reorder vs. FCFS)**
  - No. of Bank Conflicts
  - No. of column accesses per row access
  - Latency

- Effect of BIU queue depth on performance
Simulation Results
Rambus DRAM (RDRAM) – 256 Mbit per chip

Specifications
- DRAM Frequency 800 MHz
- CPU Frequency 2000 MHz
- Instruction Count - 500000000
- Channel Count – 1
- Mapping Policy – Burger Base

BURGER BASE MAP
256 Mbit = 512 rows x 32 banks x 128 col. X 16 bytes/col
16 ranks ➔ 512 MByte

31..29  28 27…..20  19 18…..15  14…..11  10 9…..4  3…..0
| | | | | | | |
unused  row id  bank id  rank id  col. id  16B pkt
FCFS vs REORDER (RDRAM)

Percentage Reduction of Bank Conflicts = 10.6673 %
FCFS vs REORDER (RDRAM)

COLUMN ACCESS PER ROW ACCESS
FOR RDRAM

<table>
<thead>
<tr>
<th>NO. OF COLUMN ACCESSES</th>
<th>FCFS</th>
<th>REORDER</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>25000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>30000</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

NO. OF OCCURRENCE
FCFS (RDRAM)

NO. OF INSTRUCTIONS VS LATENCY

LATENCY

NO. OF INSTRUCTIONS
FCFS vs REORDER (RDRAM)

COMPARISON OF FCFS AND REORDER SCHEMES FOR RDRAM

<table>
<thead>
<tr>
<th>Latency Intervals</th>
<th>FCFS</th>
<th>REORDER</th>
</tr>
</thead>
<tbody>
<tr>
<td>197-206</td>
<td>1900</td>
<td>2000</td>
</tr>
<tr>
<td>207-216</td>
<td>1500</td>
<td>1700</td>
</tr>
<tr>
<td>217-226</td>
<td>1100</td>
<td>1300</td>
</tr>
</tbody>
</table>
RIFF vs REORDER (RDRAM)

Percentage Reduction of Bank Conflicts = 10.62848 %

NO. OF CONFLICTS VS DRAM CYCLES FOR RDRAM

Percentage Reduction of Bank Conflicts = 10.62848 %
COMPARISION OF RIFF AND REORDER SCHEMES FOR RDRAM

<table>
<thead>
<tr>
<th>LATENCY INTERVALS</th>
<th>NO. OF INSTRUCTIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>197-206</td>
<td>RIFF: 1200</td>
</tr>
<tr>
<td></td>
<td>REORDER: 1900</td>
</tr>
<tr>
<td>207-216</td>
<td>RIFF: 900</td>
</tr>
<tr>
<td></td>
<td>REORDER: 1400</td>
</tr>
<tr>
<td>217-226</td>
<td>RIFF: 600</td>
</tr>
<tr>
<td></td>
<td>REORDER: 1100</td>
</tr>
</tbody>
</table>
Simulation Results

Synchronous DRAM (SDRAM)

Specifications

- DRAM Frequency 800 MHz
- CPU Frequency 2000 MHz
- Instruction Count - 500000000
- Channel Count – 1
- Mapping Policy – SDRAM Base

**SDRAM BASE MAP**

<table>
<thead>
<tr>
<th>unused</th>
<th>rank id</th>
<th>row id</th>
<th>bank id</th>
<th>col. Id</th>
<th>16B pkt</th>
</tr>
</thead>
<tbody>
<tr>
<td>31</td>
<td>30</td>
<td>29</td>
<td>28</td>
<td>27</td>
<td>26.....16</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>15</td>
</tr>
</tbody>
</table>

Ranks = 4  
Banks = 4  
Rows = 4096  
Columns = 1024
Comparision of FCFS and Reorder Schemes for SDRAM

Latency Intervals

- 187-196
- 197-206
- 207-216

No. of Instructions

- FCFS
- Reorder
Percentage Reduction of Bank Conflicts = 19.95362%
DRAM Topology

SDRAM

RDRAM
REORDER Scheme
Performance Analysis – Varying BIU Queue Depth

EFFECT OF CHANGING QUEUE DEPTH FROM 16 TO 32,48,64 ON LATENCY

LATENCY

DIFF. IN NO. OF INSTRUC

142
149
153

QUEUE DEPTH - 32
QUEUE DEPTH - 48
QUEUE DEPTH - 64
REORDER Scheme
Performance Analysis – Varying BIU Queue Depth

EFFECT OF CHANGING QUEUE DEPTH FROM
16 TO 32,48,64 ON BANK HITS

% INCR IN BANK HITS

QUEUE DEPTH - 32
QUEUE DEPTH - 48
QUEUE DEPTH - 64

DRAM CYCLES
REORDER Scheme
Performance Analysis – Varying BIU Queue Depth

EFFECT OF CHANGING QUEUE DEPTH ON CAS PER RAS COUNT

<table>
<thead>
<tr>
<th>QUEUE DEPTH</th>
<th>NO. OF OCCURRENCES</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>COL ACCESSES - 72</td>
</tr>
<tr>
<td>32</td>
<td>COL ACCESSES - 66</td>
</tr>
<tr>
<td>48</td>
<td>COL ACCESSES - 68</td>
</tr>
</tbody>
</table>
Applications – Where is reordering important?

Streaming Data:
- high spatial locality, no temporal locality
- inefficiency of dynamic caching
- access order sensitive performance

Memory Speed Bottleneck

Applications involving Streaming Computations
Design for access scheduling for streams

Stream Memory Controller
Conclusions

DRAM accesses – not really random, access order sensitive performance

Reorder Scheme

• Performance – Reduction in bank conflicts
  • RDRAM - 10.7 %
  • SDRAM - 19.9 %
  • Performance increases with increase in queue depth

• Scope for Improvement
  • Override writes
  • Priority Reordering
## Ingredients of the project

<table>
<thead>
<tr>
<th>Item</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hours at the library</td>
<td>12</td>
</tr>
<tr>
<td>Pencils</td>
<td>8</td>
</tr>
<tr>
<td>Hours of coding</td>
<td>25</td>
</tr>
<tr>
<td>Hours of simulation</td>
<td>70</td>
</tr>
<tr>
<td>Headache Pills</td>
<td>7</td>
</tr>
<tr>
<td>Cups of Starbucks Chai</td>
<td>Countless</td>
</tr>
<tr>
<td>Heated arguments</td>
<td>9</td>
</tr>
<tr>
<td>Sleepless nights</td>
<td>3</td>
</tr>
<tr>
<td>Soda</td>
<td>20</td>
</tr>
</tbody>
</table>