A Low-Cost Memory Architecture For PCI-Based Interactive Ray Casting

Michael Doggett, Michael Meißner
WSI/GRIS
University of Tübingen, Germany

Urs Kanus
DD&T GmbH
Reutlingen, Germany

Outline

Introduction
Memory Interface
PCI-Board
Results
Conclusion
**Introduction**

- **GOAL: Interactive Volume Rendering**
  - High quality images
  - Modest costs (PC-Based)
  - Flexible sampling rate
  - Color
  - Parallel and Perspective projections

---

**VIZARD I:**

- G. Knittel 1997
- **Advantages**
  - Up to 7 fps for 256³ datasets
  - Stereo and parallel projection
  - Cheap and portable
- **Disadvantages**
  - Lossy Compression
  - Pre-shading and Pre-classification
Introduction

**VIZARD II:**
- Independent ray casting accelerator
- Complete rendering pipeline on board
- Parallel and perspective projection
- Cut planes and interactive classification support
- Interactive frame-rates for $256^3$ voxels
- **Input:** Viewing Parameters
- **Output:** Images

Memory Performance

*Why improve Memory Performance?*
- The user wants it faster and bigger
- Frame rate is dominated by memory access time
- Higher speed and storage for larger datasets - $512^3$
- Improve caching - VIZARD I - Ray coherence cache
Introduction

Sub-Cubes - Cubic Addressing

- Lichtermann95 - Sub-Cubes
- de Boer96 - Sub-Cubes
- Osborne97 EM-Cube, Block Skewed Memory
- Vettermann99, 8 Parallel Memories with Cubic Addressing

Outline

- Introduction
- Memory Interface
- PCI-Board
- Results
- Conclusion
Memory Interface

**Issues**

- 8 voxel neighbourhood for tri-linear interpolation
- Random access
- Modern memories
  - RDRAM
  - SDRAM, Random column read per cycle within page of same bank - Cache
Memory Interface

**Organisation**

- 8 Memories to read 8 voxel neighbourhood in one cycle
- Addressing calculation based on sub-cubes ($8^3$)
  - Cubic Address = Sub-Cube + Voxel Address
  - $= \{x,y,z\}/8 + \{x,y,z\} \mod 8$

Memory Structures

64 Memories
- Maintain 8 active caches
- Activate all 56 neighbouring caches
- Impractical

8 Parallel Memories
- Feasible
- Stalls at boundaries for row activate and precharge

Prefetching
SDRAM State Diagram

- **PREcharge**
- **ACTivate**
- **READ**
- **Same Row?**
- **Precharged?**

Memory State

Manual Input

Automatic Sequence

Controller Test

Controller State Change

Test Reads in Previous Bank

If Bank Precharged

Test Reads in same Bank

If Same Row

If  Bank

PREcharge

ACTivate

READ

Same Row?

Precharged?
SDRAM State Diagram

- PREcharge
- ACTivate
- READ
- Same Row?
- Precharged?

Memory Interface

Cache reloads and pipeline stalls

- 1 - 1 cache
- 2 - 2 caches - 2 reloads
- 3 - 1 cache - 2 reloads
- 4 - 2 caches - 2 reloads
- 5 - 1 cache - 2 reloads

4 pipeline stalls
### Prefetching

<table>
<thead>
<tr>
<th>State</th>
<th>Address FIFO</th>
<th>Memory</th>
<th>Voxel FIFO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial</td>
<td>Address FIFO</td>
<td>Memory</td>
<td>Voxel FIFO</td>
</tr>
<tr>
<td>Miss -&gt; Stall</td>
<td>Stall</td>
<td>Stall</td>
<td>Stall</td>
</tr>
<tr>
<td>Fill Voxel FIFOs</td>
<td>Stall</td>
<td>Stall</td>
<td>Stall</td>
</tr>
<tr>
<td>Row Active</td>
<td>Stall</td>
<td>Stall</td>
<td>Stall</td>
</tr>
<tr>
<td>Miss -&gt; No Stall</td>
<td>Stall</td>
<td>Stall</td>
<td>Stall</td>
</tr>
<tr>
<td>Continue</td>
<td>Stall</td>
<td>Stall</td>
<td>Stall</td>
</tr>
</tbody>
</table>

---

<table>
<thead>
<tr>
<th>State</th>
<th>Address FIFO</th>
<th>Memory</th>
<th>Voxel FIFO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial</td>
<td>Address FIFO</td>
<td>Memory</td>
<td>Voxel FIFO</td>
</tr>
<tr>
<td>Miss -&gt; Stall</td>
<td>Stall</td>
<td>Stall</td>
<td>Stall</td>
</tr>
<tr>
<td>Fill Voxel FIFOs</td>
<td>Stall</td>
<td>Stall</td>
<td>Stall</td>
</tr>
<tr>
<td>Row Active</td>
<td>Stall</td>
<td>Stall</td>
<td>Stall</td>
</tr>
<tr>
<td>Miss -&gt; No Stall</td>
<td>Stall</td>
<td>Stall</td>
<td>Stall</td>
</tr>
<tr>
<td>Continue</td>
<td>Stall</td>
<td>Stall</td>
<td>Stall</td>
</tr>
</tbody>
</table>
Prefetching

- Initial
- Miss -> Stall
- Fill Voxel FIFOs
- Row Active
- Miss -> No Stall
- Continue

M. Doggett, M. Meißer, U. Kanus
University of Tübingen, WSI/GRIS
### Prefetching

**Initial**

- Miss -> Stall
- Fill Voxel FIFOs
- Row Active
- Miss -> No Stall
- Continue

<table>
<thead>
<tr>
<th>Address FIFO</th>
<th>Memory</th>
<th>Voxel FIFO</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Cache Miss
- FIFO Empty
- Stall

- Initial
- Miss -> Stall
- Fill Voxel FIFOs
- Row Active
- Miss -> No Stall
- Continue

<table>
<thead>
<tr>
<th>Address FIFO</th>
<th>Memory</th>
<th>Voxel FIFO</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Cache Miss
- FIFO Empty
- Stall
Prefetching

Address FIFO | Memory | Voxel FIFO

<table>
<thead>
<tr>
<th>Address FIFO</th>
<th>Stall</th>
<th>Address FIFO</th>
<th>Stall</th>
<th>Address FIFO</th>
<th>Stall</th>
<th>Address FIFO</th>
<th>Stall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory</td>
<td></td>
<td>Voxel FIFO</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FIFO Empty</td>
<td></td>
<td>FIFO Empty</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1 - 1 cache
2 - 2 caches - 2 reloads
3 - 1 cache - 2 reloads - No Stall
4 - 2 caches - 2 reloads
5 - 1 cache - 2 reloads - No Stall

Only 2 pipeline stalls
Memory Structures

4 DIMMs
- Off the shelf
- Cheap
- Upgradeable
- Large capacity (up to 1 GB)
- 4 addresses ➔ Data replication

Memory Interface

Disadvantage - Data replication

Memory Modules

Trilinear Interpolator

Sample
Outline

Introduction
Memory Interface
PCI-Board
Results
Conclusion

PCI Board

Xilinx XCV 10000 - 1M gates

DSP - Ray Set Up and Bus Control

LUT

PCIPCI Interface

DIMMDIMM

Xilinx Virtex

DIMMDIMM

Xilinx - XCV 10000 - 1M gates
Outline

Introduction
Memory Interface
PCI-Board
Results
Conclusion

Results

Foot - 256^3
Arteries - 256^3
Statue 1 - 341^2 x 364 (original data 341^2 x 91)
Statue 2 - Double sampling in view direction

Software Simulation using SDRAM state model and NEC 100MHz SDRAM timings
Results

VHDL simulation

- Initial image result from mixed structural and behavioural VHDL simulations of VIZARD II
Conclusion

- Improved memory performance by combining Cubic Addressing and Buffering
- DIMMs are a Cheap, Upgradeable, Large capacity Memory solution
- Memory Interface simulated and synthesised in VHDL for Xilinx technology
- Perspective projection high quality images
- First Prototype expected fall 1999
www.volsight.de