SystemC-based Design Space Exploration of a 3D Graphics Acceleration SoC for Consumer Electronics

Tse-Chen Yeh¹, Tsung-Yu Ho¹, Hung-Yu Chen¹, and Ing-Jer Huang¹

¹ Department of Computer Science and Engineering, National Sun Yat-sen University, Kaohsiung, 804, Taiwan
{tcyeh, tyho, hychen}@esl.cse.nsysu.edu.tw
ijhuang@cse.nsysu.edu.tw

Presenter Chi-Tsai Yeh
Outline

- Abstract
- Motivation
- Related work
- 3D graphics acceleration SoC platform
- The hardware/ software configurations
- Design space exploration
- Conclusion
Abstract

In order to solve the system performance bottleneck of a 3D graphics acceleration SoC, we exploit design space exploration on performance evaluation and benchmark characteristics using SystemC. We find out the bottleneck according to the simulation results of 9 hardware/software configurations and find out the tradeoffs between different configurations. The performance issues of SoC have been explored under the low-cost constraints, such as cache size effect, hardware accelerations and memory traffic. In conclusions, we provide the performance/cost tradeoffs and 3D graphics benchmark features for designing a 3D graphics SoC.
Outline

- Abstract
- **Motivation**
- Related work
- 3D graphics acceleration SoC platform
- The hardware/software configurations
- Design space exploration
- Conclusion
Motivation

What’s the problem?

- Consumer electronics are cost-sensitive, and customers demand higher performance with lower price. A developer needs the product to meet the performance under the cost constraint.
  - Higher performance
  - Lower price

Proposed approach

- Design space exploration using SystemC evaluates the system performance
Design Flow

1. Pure software
2. Additional hardware or architecture refinement
3. Requirement meet?
4. Yes
5. No
6. Bottleneck appears
7. adapt HW/ SW configuration
Outline

- Abstract
- Motivation
- Related work
- 3D graphics acceleration SoC platform
- The hardware/software configurations
- Design space exploration
- Conclusion
CoWare™ Platform Architect

- Provides the hardware/software integration and simulation platform using systemc
- Software written by C/C++ can be translated effortlessly into systemc hardware model because systemc is a C++ class library
- Provides IP library
- Cycle-accurate transactional bus simulator (TBS)
A tile-based 3D graphics pipeline

Geometry Processing

- Viewing Transform
- Lighting
- Perspective Transform
- Tile Divide

Vertex information

Raster Processing

- Scan Conversion
- Shading
- Z-Test
- Frame buffer
Outline

- Abstract
- Motivation
- Related work
- 3D graphics acceleration SoC platform
- The hardware/software configurations
- Design space exploration
- Conclusion
System functional block of 3D graphics acceleration SoC

- ARM926EJ-S
- Geometry Module
- Tile Divider
- Raster Module
- External Memory (SDRAM)
- MEM Controller
- WO-DMA
- Display Engine
- BI
- BI

Vertex information, temporary data, frame buffer and Z buffer

Pure software environment with SDRAM latency

Receives the triangle data from GM and generate tile list to RM

3D Graphics Engine

Clear buffer

Display result frame

SystemC HW model

2007/12/19
Outline

- Abstract
- Motivation
- Related work
- 3D graphics acceleration SoC platform
- The hardware/software configurations
- Design space exploration
- Conclusion
System Configurations

- Total 9 hardware/software configurations will be introduced for our design space exploration.
- Four of these configurations are used to evaluate software acceleration, and the others are used to explore different hardware solutions.
- Three benchmarks for 3D graphics:
  - Teapot -- more raster processing
  - Helicopter -- more geometric computations
  - Elephant -- the most complexity
## Features of three 3D graphics benchmarks

<table>
<thead>
<tr>
<th></th>
<th>Teapot</th>
<th>Helicopter</th>
<th>Elephant</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total vertices number</td>
<td>12,288</td>
<td>15,414</td>
<td>87,840</td>
</tr>
<tr>
<td>Processed tiles</td>
<td>135</td>
<td>64</td>
<td>141</td>
</tr>
</tbody>
</table>
# Hardware/Software Configurations for Design Space Exploration

<table>
<thead>
<tr>
<th>Configuration Name</th>
<th>Function</th>
<th>Bus Interface</th>
<th>SDRAM Latency</th>
<th>Cache</th>
<th>Display Engine</th>
<th>Geometry FIFO</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Clear Buffer</td>
<td>Geometry</td>
<td>Raster</td>
<td>Transfer Mode</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pSW (ADS)</td>
<td>SW</td>
<td>SW</td>
<td>SW</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>pSW (ADS S)</td>
<td>SW</td>
<td>SW</td>
<td>SW</td>
<td>N/A</td>
<td>N/A</td>
<td>ON</td>
</tr>
<tr>
<td>pSWD (PA)</td>
<td>SW</td>
<td>SW</td>
<td>SW</td>
<td>N/A</td>
<td>ON</td>
<td>N/A</td>
</tr>
<tr>
<td>pSWD (PA S)</td>
<td>SW</td>
<td>SW</td>
<td>SW</td>
<td>N/A</td>
<td>ON</td>
<td>ON</td>
</tr>
<tr>
<td>GED_s</td>
<td>SW</td>
<td>HW</td>
<td>HW</td>
<td>Single</td>
<td>ON</td>
<td>N/A</td>
</tr>
<tr>
<td>GED_b</td>
<td>SW</td>
<td>HW</td>
<td>HW</td>
<td>Burst</td>
<td>ON</td>
<td>N/A</td>
</tr>
<tr>
<td>GDGED_b</td>
<td>GDMA</td>
<td>HW</td>
<td>HW</td>
<td>Burst</td>
<td>ON</td>
<td>N/A</td>
</tr>
<tr>
<td>WDGED_b</td>
<td>WDMA</td>
<td>HW</td>
<td>HW</td>
<td>Burst</td>
<td>ON</td>
<td>N/A</td>
</tr>
<tr>
<td>WDGED_Fb</td>
<td>WDMA</td>
<td>HW</td>
<td>HW</td>
<td>Burst</td>
<td>ON</td>
<td>N/A</td>
</tr>
</tbody>
</table>
Cache Size Optimization without Hardware Acceleration

- Pure software using SDRAM on AMBA bus
- Pure software using SRAM on ideal bus
- Performance degradation cause cache size is too small
- Bus & SDRAM overhead
- Minimum cache size of ARM926 IP @ PA
# Cache size optimization on software execution

<table>
<thead>
<tr>
<th></th>
<th>pSW (ADS) (cycles)</th>
<th>pSW (ADS$4) (cycles)</th>
<th>Improvement (ADS)</th>
<th>pSW (PA) (cycles)</th>
<th>pSW (PA$4) (cycles)</th>
<th>Improvement (PA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clear Buffer</td>
<td>2,935,822</td>
<td>2,167,247</td>
<td>26.2%</td>
<td>9,003,353</td>
<td>5,778,533</td>
<td>35.8%</td>
</tr>
<tr>
<td>Geometry</td>
<td>133,002,461</td>
<td>66,931,465</td>
<td>40.1%</td>
<td>396,033,200</td>
<td>209,441,410</td>
<td>47.1%</td>
</tr>
<tr>
<td>Raster</td>
<td>139,867,002</td>
<td>97,181,194</td>
<td>30.5%</td>
<td>460,224,000</td>
<td>284,340,000</td>
<td>38.2%</td>
</tr>
<tr>
<td>Total</td>
<td>412,376,118</td>
<td>260,850,843</td>
<td>36.7%</td>
<td>865,260,553</td>
<td>499,559,943</td>
<td>42.3%</td>
</tr>
</tbody>
</table>
# Efficiency of Bus Transfer Mode

<table>
<thead>
<tr>
<th>Burst transfer</th>
<th>Teapot</th>
<th>Helicopter</th>
<th>Elephant</th>
</tr>
</thead>
<tbody>
<tr>
<td>Geometry Speedup</td>
<td>36%</td>
<td>52%</td>
<td>49.7%</td>
</tr>
<tr>
<td>Raster Speedup</td>
<td>33.1%</td>
<td>47.4%</td>
<td>54.7%</td>
</tr>
</tbody>
</table>

**Single transfer (GED_s) vs. Burst transfer (GED_b)**
Performance Bottleneck on Clear Buffer

- Clear Buffer: 1.0%
- Geometry: 45.7%
- Rasterization: 53.1%

Execution time reduction: 98.54%

Pure SW

H/W acceleration

Geometric: 3.4%
Rasterization: 9.7%
Clear Buffer: 86.9%
### Performance improvement on Clear Buffer

<table>
<thead>
<tr>
<th></th>
<th>Teapot</th>
<th>Helicopter</th>
<th>Elephant</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Generic DMA (GDGED_b)</strong></td>
<td>50.5%</td>
<td>41.77%</td>
<td>41.40%</td>
</tr>
<tr>
<td><strong>Write-only DMA (WDGED_b)</strong></td>
<td>90.76%</td>
<td>89.69%</td>
<td>89.20%</td>
</tr>
</tbody>
</table>

*Compare with GED_b*
## Performance/cost tradeoffs

<table>
<thead>
<tr>
<th>hardware/software configuration</th>
<th>Performance improvement</th>
<th>hardware cost (gate count)[5,6]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Teapot</td>
<td>Helicopter</td>
</tr>
<tr>
<td>Software only (pSWD(PA))</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>I/D cache (pSWD(PD$))</td>
<td>42.60%</td>
<td>72.51%</td>
</tr>
<tr>
<td>GE (GED_b)</td>
<td>98.54%</td>
<td>98.81%</td>
</tr>
<tr>
<td>GE + WO-DMA (WDGED_b)</td>
<td>99.69%</td>
<td>99.74%</td>
</tr>
<tr>
<td>GE + WO-DMA+ Ping-pong buffer (WDGED_Fb)</td>
<td>99.80%</td>
<td>99.75%</td>
</tr>
</tbody>
</table>
Characteristics of 3D Graphics Benchmarks

(a) Geometry characteristics

(b) Rasterization characteristics
Outline

- Abstract
- Motivation
- Related work
- 3D graphics acceleration SoC platform
- The hardware/software configurations
- Design space exploration
- Conclusion
Conclusion(1)

In this paper, we use the design space exploration for performance evaluation and the performance/cost tradeoffs by exploiting 9 hardware/software configurations. And we also characterize the benchmark features for advanced survey on 3D graphics processing. The performance evaluation and benchmark characteristics can give many benefits and references for system designers to determine their system architecture and hardware/software configurations.
Conclusion(2)

Notice that memory traffic is still the performance bottleneck of a low-cost SoC which applied on 3D graphics rendering. And texture mapping, memory allocation, comparisons of alternative architecture modeling, power/ performance analysis, and multi-layer bus architecture for 3D graphics acceleration SoC will become open issues in our future works.