Hot3D Presentations

General Chairs
Michael Doggett
Anselmo Lastra

Program Chairs
Mark Harris
David Luebke

Papers Chairs
Michael McCool
Tomas Akenine-Möller
## Contents

**ORAD’s DVG : Solutions for Scalable Graphics Clusters**  
Guillaume Otrage, ORAD  
1

**Mali Graphics IP Cores**  
Edvard Sørgård, Falanx Microsystems  
13

**Bitboys G40 Embedded Graphics Processor**  
Petri Nordlund, Bitboys Oy  
19
Orad’s DVG: solutions for scalable graphics clusters

Scalable architecture

e.g.: compositing 3 cards to one channel in HP workstation
The DVG architecture

- PC cluster solution for scalable performance on multiple or SINGLE displays.
  - M nodes drive N displays, M & N are any numbers, M>N
- At the heart of the DVG technology is the DVG board, a highly programmable (FPGA) PCI card.
  => achieving genlocking AND combining.
- The DVG board directly interacts with the graphical board allowing fully linear scaling of performance
- A dedicated patent-pending pixel bus “DVG-bus” interconnects the DVG units so no graphics, CPU or motherboard resources are sacrificed
- DVG board occupies space of 1 to 2 PCI slots depending on version.

Combining (a.k.a. “chaining”)

The DVG supports several different combining modes:
- Sample division (for Anti-Aliasing)
- Time division
- Image division
- Eye division etc...

Composition is done on board, no need to move pixels back into memory with associated overhead in performance and # of CPUs.

Various chaining modes can be mixed

Modes can be switched with no change of wiring

Combining configuration can be chosen according to the critical system/data-base resource
More on combining: Anti-aliasing

- Anti-aliasing is critical for image quality
- One DVG rendering unit can implement for instance 4-sample anti-aliasing but with a certain performance cost (fill rate)
- Multiple DVG rendering units connected together (see below) can do up to 16-sample anti-aliasing with same performance as a 4 sample rendering
- This mode does not add delay

More on combining: Time-division chaining

- This mode scales performance linearly with the number of units in the chain, without any limit
- With N units in the chain and 60Hz clock each node generates 60/N images per second.
- Therefore each rendering unit has N field times to generate its graphics
- So it can render graphics N times more complex than a single unit could
- The final output of course is still a 60 images per second display
- This mode generates latency of N-1 frame times
More on combining: Imaging division chaining

- Each rendering unit is “responsible” for a sub-region of the final output image
- This mode is better suited for applications where performance is limited by pixel fill such as high resolution formats
- No added latency
- Special dynamic load balance algorithms:
  - “Quadrant division” with a dynamic rendering time criterion
  - “Interleaved division”
- The “Quadrant division” method can also improve the geometry processing using culling

Imaging division chaining cont’d

- Image division effectively increases available texture memory, even more so as it is combined with AA chaining:
  - eg: a 1600x1200 renderer at 6 samples uses 95Mb of texture memory for buffering, remaining only \(33\text{Mb} \) (assuming 128 Mb total)
- While 1024 x 768 rendering screen at 2 samples uses 15Mb of texture leaving \(133 \text{ Mb available} \)!

=> Image division and anti-aliasing chaining increase available texture memory for graphic textures
Split image division chaining

- Each unit renders fragment of the scene. Combiner creates output image bigger than input components.
- Both vertical and horizontal image division is allowed
- Application can use view culling to gain geometry rate
- Static load-balanced gain on pixel fill rate

Interleaved image division chaining

- Each unit renders full scene, but in window “squeezed” horizontally
- Each window has projection matrix shifted by a subpixel. Combiner interleaves pixels to produce output image with higher resolution.
- No gain on geometry rate but dynamic load-balanced gain on pixel fill rate
- Cannot use antialiasing of graphic card until programmable sample locations are available.
Dynamic image division chaining

- Each unit renders fragment of the scene in viewport smaller than full window. The rest of window is filled with black. Combiner adds images.
- Application can use view culling to gain geometry rate.
- Viewports can be resized on the fly, so application can do dynamic load balancing for pixel fill rate.
- Overhead time (e.g. 'swapbuffers') is bigger (because each unit renders in full window).

More on combining:
Eye division chaining

- For active stereo (96,110,120hz), left/right eye division: linear performance increase
- The first chosen method in CAVE, VRs, as it is one of the most efficient.
- No added delay
More on combining: Scene division chaining

- The 3D objects in the scene are split between the different rendering units.
- Each pixel’s distance to the camera (its Z value) is transmitted between rendering units.
- Based on this information, the DVG boards in the chain correctly composite the final image.
- Performance scales linearly with the number of rendering units.
- Differently from the other methods, this method not only increases total performance, but also effectively increases graphics resources such as texture memory, beyond the capability of a single rendering unit.
- No added delay but no access to FSAA.

More on combining: Volume rendering division

- Many visualization applications, primarily in oil & gas and medical imaging are based on volume, not polygonal rendering.
- The process can be implemented on COTS graphics hardware.
- The cluster linearly increases performance.
- No added delay.
DVG Formats

- The DVG supports all formats up to output pixel frequency of **300 MHz (!!!)**
- Supported VESA and other standards:
  - All SDI and HD video formats
  - All 800x600, 1024x768, 1280x960, 1280x1024 formats
  - 1280x1024 120Hz (stereo)
  - 1600x1200 (60-85Hz)
  - 1792x1344 (60, 75Hz)
  - 1856x1392 (60, 75Hz)
  - 1920x1440 (60, 75Hz)
  - 2048x1536 60Hz

Software certification and porting

- Easy porting of generic (Vega Prime, Performer, Amira, Opticore Opus, Virtools...) and customized rendering applications, authoring, assembly and data base manipulation tools
- 3 methods for porting:
  - **DVGlib** – a library used for the DVG system administration providing DVG specific functionality like setting up chain configuration, format, as well as synchronized swap buffer.
  - **DVG wrapper** – intercepts OpenGL calls in order to configure rendering for compositing
  - **DVG wire** – allows non-cluster applications (CATIA DMU, EDS/UGS Viz Mockup, PTC DV mockup...) to run in distributed mode on a DVG-based cluster
- Successful porting processes have already been implemented by customers like British Aerospace, Lockheed Martin and others (typical duration – few hours)
- The DVG supports both Windows and Linux based rendering codes
- Orad offers porting assistance services by a dedicated team
- The OpenGL driver is the unmodified driver supplied by the graphics card maker and thus provides the most up-to-date extensions and optimizations as they are released
More DVG features

- Hardware based image post – processing:
  - NVG / Flir “look”
  - Chromakey (for augmented reality...)

- Multiple video insertions (mapped on a polygon or as overlay)
  - Optional
  - Instructor’s video
  - Cockpit monitors
  - Collaborative session
  - Augmented reality

More DVG features

- VIZ CLUSTER MANAGEMENT ("VCM"):
  - Encompass all the HW and SW technologies designed by Orad for visualisation cluster management:
    - Multiple renderers can be shared by a group of users over a network. Depending on the day's use, resources are allocated across the network to each user (eg: a 4 channel with 2 renderers on each becomes a 2 channel with 4 renderers on each)
    - Cluster Permanent Availability “CPA” software: any broken renderer in a channel can automatically be by-passed so that for instance the channel has lower AA but continues working
  - VCN is managed by the DVG service software which also handles the reconfiguration of combining depending on application, as well as selection of formats. Manages VCN from a logical/SW standpoint
DVG applications

- CAVEs, Workbench, Flight simulators...
- Civil and military simulations
- Mission planning
- Urban planning
- Car design
- Car driving simulation
- Interactive walk through
- Theme parks
- Architectural design
- Scientific/Medical visualization
- Collaborative Engineering
- Museums, Planetariums and Cultural Centers
- Hazard Perception / Disaster Management
- Oil & Gas explorations
- Homeland Security
- Augmented reality

ANNEX: miscellaneous supporting examples
Screen shots from an integration in HP’s XW8000 workstation, Orad’s privileged partner for workstations.

Form-factors : DVG VR-X (1/2)

Form-factors : DVG10 VR (2/2)

Orad DVG10 VR backpanel. Orad’s offering for integrated rack-mounted PCs.

<table>
<thead>
<tr>
<th>Port</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>Reference PMO 20MBPS/50MBPS</td>
</tr>
<tr>
<td>12</td>
<td>Reference PMO 20MBPS/50MBPS</td>
</tr>
<tr>
<td>21</td>
<td>Reference PMO 20MBPS/50MBPS</td>
</tr>
<tr>
<td>22</td>
<td>Reference PMO 20MBPS/50MBPS</td>
</tr>
<tr>
<td>23</td>
<td>Reference PMO 20MBPS/50MBPS</td>
</tr>
<tr>
<td>24</td>
<td>Reference PMO 20MBPS/50MBPS</td>
</tr>
<tr>
<td>25</td>
<td>Reference PMO 20MBPS/50MBPS</td>
</tr>
<tr>
<td>26</td>
<td>Reference PMO 20MBPS/50MBPS</td>
</tr>
<tr>
<td>27</td>
<td>Reference PMO 20MBPS/50MBPS</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Port</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>Reference PMO 20MBPS/50MBPS</td>
</tr>
<tr>
<td>12</td>
<td>Reference PMO 20MBPS/50MBPS</td>
</tr>
<tr>
<td>21</td>
<td>Reference PMO 20MBPS/50MBPS</td>
</tr>
<tr>
<td>22</td>
<td>Reference PMO 20MBPS/50MBPS</td>
</tr>
<tr>
<td>23</td>
<td>Reference PMO 20MBPS/50MBPS</td>
</tr>
<tr>
<td>24</td>
<td>Reference PMO 20MBPS/50MBPS</td>
</tr>
<tr>
<td>25</td>
<td>Reference PMO 20MBPS/50MBPS</td>
</tr>
<tr>
<td>26</td>
<td>Reference PMO 20MBPS/50MBPS</td>
</tr>
<tr>
<td>27</td>
<td>Reference PMO 20MBPS/50MBPS</td>
</tr>
</tbody>
</table>

Graphics Hardware 2004
Hot3D presentations
Company Overview

- Design and license silicon graphics IP cores targeted at mobile phones and system-on-chip
- Core Competencies
  - Computer Graphics Architectures and Algorithms
  - Hardware Description Languages and Tools
  - Software Design and Development
- Norway / US
  - Trondheim – the technology capitol of Norway
  - Falanx Inc.
- Zoran first licensee
Overview

• Mali Graphics IP Cores
  – First implementation of Falanx' architecture
  – Scales with the OpenGL®ES road-map
  – 4X Full Scene Anti-Aliasing Standard
    • Up to 100-200MPix / s dependent on texturing, etc.
    • No measurable decrease in performance or increase in bandwidth usage.
  – 16X Full Scene Anti-Aliasing Option
Image Quality

- Anti-Aliasing Important For IQ
  - Small displays can look brilliant
- Several algorithms, but
  - Unacceptable increase in memory BW / power
  - Unacceptable performance hits
- No compromise

Block Diagram

![Block Diagram of Graphics Hardware Components](image-url)
Current Products

<table>
<thead>
<tr>
<th>Mali100+G</th>
<th>Mali50+G</th>
<th>Mali100</th>
<th>Mali50</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max Clock</td>
<td>180MHz</td>
<td>180MHz</td>
<td>180MHz</td>
</tr>
<tr>
<td>M Pix / s</td>
<td>180</td>
<td>90</td>
<td>180</td>
</tr>
<tr>
<td>M Tri / s</td>
<td>5</td>
<td>2.5</td>
<td>CPU</td>
</tr>
<tr>
<td>Total Area</td>
<td>5 mm²</td>
<td>4 mm²</td>
<td>3.5 mm²</td>
</tr>
</tbody>
</table>

- Performance given with 4X FSAA Mode
  - Bilinear single texture
- Area given with Free Artisan TSMC 130nm library
  - Includes scan, clock gating and SRAMs

Rich Feature Set

Key Features
- 4X FSAA Standard Operation
- 16X FSAA at your request
- Video Primitives Acceleration
- Texture Compression (FLXTC)
- OpenGL ES Feature Set and more

System features
- 16 / 32 bit frame buffer
- Max. Resolution 2048x2048
- Autonomous Frame Rendering
- Memory Management Unit

Other Hardware Accelerated Features - High lights
- Points / Lines / Triangles / Quads
- Flat / Gouraud Shading
- Point / Bi-linear / Tri-linear Texturing
- Multi texturing
- Auto Mip Map Generation
- Dot3 Bump Mapping
- Flexible Texture Input formats
- Aggressive Z Tests
- Triangle Setup
- 2D / Point / JSR184 Sprites
- Anti-aliased font rendering
- Bitbit / ROP3/4
- Vertex Shader 2.0
- 4-bit Stencil Buffers
- Specular Color / Color Sum
- Render To Texture w/ AA
- Matrix Palette Skinning
- DCT / iDCT
Silicon IP Cores

• Soft Cores are Challenging because of
  – Tool Chains, Coding and design styles
    • Requires extensive chip design and HDL knowledge
  – Different System-on-Chip Architectures and OSs needs special attention
    • For maximum performance software and hardware must be designed together
    • Different memory controllers, bus latencies and bus arbiter implementations affects the graphics system

• FHDL – Falanx HDL
  – Compiles to different targets (Verilog, VHDL, ++)
  – Verified and proven through several different HDL simulators, synthesis tools, back-end flow
  – Enables quick and painless transition to new tools, embedding of “special requests” and coding styles

• Software Development
  – ANSI C or die
  – ARM and OS optimizations give an edge

• Falanx IP cores:
  – Affects the SoC design as little as possible
  – Are as pluggable, re-usable and scalable as possible
Falanx Microsystems

Demonstration
Bitboys G40
Embedded graphics processor

Petri Nordlund
CTO, Bitboys

Brief history of embedded graphics hardware

- The early contenders
  - Bitboys G10: SVG Tiny vector graphics acceleration
  - Other propriety, non-standard 3D graphics hardware

- The standards are ratified (OpenGL® ES 1.0)
  - ATI Imageon, NVIDIA GoForce
  - Bitboys G30
  - Imagination MBX
  - Mali series from Falanx
  - Sanshin’s G-Shark

- The standards mature (OpenGL® ES 1.1)
  - Bitboys G32 and G34

- Future standards
  - Targeting programmability, OpenGL® (ES) 2.0
  - Bitboys G40
Current graphics processors

- Targeting OpenGL® ES 1.1, typical features
  - OpenGL® ES 1.1 pixel pipeline in hardware
  - 32-bit color (8-8-8-8)
  - Some form of texture decompression (2bpp or 4bpp)
  - Full-screen anti-aliasing
  - 1 pixel / clock

- Optional: Hardware transformation and lighting
  - Fixed-function or limited programmability
  - Choice of integrating hardware T&L depends heavily on target
    system – not necessarily required if CPU has floating point
    processing capability

- Design sizes (typical for all contenders)
  - <400 Kgates without hardware T&L
  - Hardware T&L adds 150-400 Kgates

G40 - Introduction

- Graphics processor IP core designed and optimized for handheld devices
  - Integrates into an SoC, connects to the system memory bus
  - Supports OCP, AMBA AHB or customer specific buses

- Targeting consumer products in 2007-2010 timeframe
  - Mobile phones (feature and smart-phones)
  - Handheld gaming devices
  - Other embedded devices (PDAs, car navigation, set-top boxes)

- 2D, 3D and vector graphics acceleration
  - Programmable, floating-point vertex shader (32-bit IEEE)
  - Programmable, floating-point pixel shader (16-bit OpenEXR)
  - Complete OpenGL® ES 1.1 pipeline in hardware

- Target content
  - Device’s user interface, games, application graphics
G40 - Main development guidelines

- Target volume market mobile phones in 2007-2010 timeframe
  - We expect 3D graphics breakthrough in mobile phones in 2006 timeframe – Japan first, then Europe, followed by US

- Industry standard content creation tools and game art will be largely based on the use of shaders
  - Don’t want to stray from this path

- Scene complexity and performance target
  - 60 FPS
  - 20-30k polygons/frame
  - QVGA or VGA display resolution
  - Depth complexity 5
  - Relatively complex pixel shaders
  - High sustained pixel fillrate

G40 - Main development guidelines (continued)

- Power consumption
  - Careful selection of features to reduce hardware size
  - Programmable architecture instead of fixed-function
  - Intelligent power management

- Process technology
  - 90 or 65 nm are used for mobile phone SoCs in this timeframe
  - 200 MHz peak clock frequency

- “Feature-proof” architecture
  - Product cycles on the embedded side are long
    - Large number of IP blocks integrated into heavy SoCs
    - Standardization takes a lot of time
    - Mobile phones are all about standards
  - Need to make a bet for which features to support → programmability provides safety
G40 – Rendering features

- 2D graphics rendering
  - BitBlts, fills, ROPs (256)
  - Small separate core for rendering bitmap-based user interfaces

- Vector graphics rendering
  - SVG Basic level feature set, targeting OpenVG
  - Anti-aliased rendering of concave and convex polygons
  - Rasterization integrated into the 3D pipeline
  - Support for linear and radial gradients
  - Arbitrary clip paths
  - 10-50x performance over software rendering

- 3D graphics
  - Transformation and lighting in hardware
  - Floating-point vertex and pixel shaders
  - Multitexturing: Four textures per pixel
  - Fully programmable architecture, no fixed-function pipeline
  - FLIPQUAD full-screen anti-aliasing
  - PACKMAN hardware texture decompression

Why vector graphics

- Very suitable mobile and handheld devices
  - Resolution independent
  - Small content size
  - High-quality anti-aliased images

- Strong customer demand for hardware accelerated vector graphics rendering

- Usage:
  - User interfaces
  - Interactive applications
  - (Streaming) cartoons
  - Greeting cards
  - Procedural texture generation for 3D games

- Software APIs
  - OpenVG from Khronos
  - SVG (Scalable Vector Graphics)
Architecture

- Rendering pipeline based on OpenGL® 2.0 shader architecture
- Fully floating-point, programmable, well integrated architecture
- Fixed function fully emulated using the programmable pipeline
- Designed from ground up to power mobile phones and other handheld devices