Overview

- System architecture
  - Rendering performance
  - GPU architecture
- Unified shader
- Memory Export
- Texture/Vertex Fetch
- HDR rendering
- Displaced subdivision surfaces
System architecture

- **CPU**: 2x 10.8 GB/s
- **Southbridge**: 2x PCIE 500MB/s
- **GPU**: 32GB/s
  - **UNIFIED MEMORY**: 22.4GB/s
  - 700MHz 128bit GDDR3
- **Northbridge**: 2x 10.8 GB/s
- **DAUGHTER DIE**: 32GB/s

Rendering performance

- **GPU to Daughter Die interface**
  - 8 pixels/clk
    - 32BPP color
    - 4 samples Z - Lossless compression
  - 16 pixels/clk – Double Z
    - 4 samples Z - Lossless compression
- **GPU**: 32GB/s
- **DAUGHTER DIE**: 32GB/s
Rendering performance

- Alpha and Z logic to EDRAM interface
  - 256GB/s
  - Color and Z - 32 samples
    - 32bit color, 24bit Z, 8bit stencil
  - Double Z - 64 samples
    - 24bit Z, 8bit stencil

GPU architecture

- DAUGHTER DIE
  - 8pix/clk, 4x MSAA, Stencil and Z test, Alpha blending
  - 256GB/s
  - 10MB EDRAM

- UNIFIED MEMORY
  - Texture/Vertex Fetch
  - Index Stream Generator
  - Rasterizer
  - Clipper
  - Primitive Setup
  - Output Buffer
  - Memory Export
  - Vertex Pipeline
  - Pixel Pipeline
  - Display Pixels
Unified Shader

- A revolutionary step in Graphics Hardware
- One hardware design that performs both Vertex and Pixel shaders
- Vertex processing power

Unified Shader

- GPU based vertex and pixel load balancing
  - Better vertex and pixel resource usage
- Union of features
  - E.g. Control flow, indexable constant, ...
- DX9 Shader Model 3.0+
Memory Export

- Shader output to a computed address
- Virtualize shader resources - multipass
- Shader debug
- Randomly update data structures from Vertex or Pixel Shader
- Scatter write

Texture/Vertex Fetch

- Shader fetch can be either:
  - Texture fetch (16 units)
    - LOD computation
    - Linear, Bi-linear, Tri-linear Filtering
    - Uses cache optimized for 2D, 3D texture data with varying pixel sizes
    - Unified texture cache
  - Vertex fetch (16 units)
    - Uses cache optimized for vertex-style data
Texture Arrays

- Generalization of 6 faced cube maps to 64 faces
- Each face is a 2D mip mapped surface
- Not volume texture
- Applications
  - Animation frames
  - Varying skins for instanced characters / objects
  - Character shadow texture flipbook animations

Texture array application:
Unique seeds for instanced shading
Texture array application:
Hundreds of instanced characters

Texture compression

- All of the old DXT formats
  - DXT1, DXT2/3, DXT4/5
- Several new formats (variations on above formats)
  - DXT3A
    - 4 bit scalar replicated into four channels in shader
  - DXT3A as 1111
    - 1 bit per channel pixel
  - DXT5A
    - 3bit selection between 2 8bit endpoints
  - DXN
    - 3Dc normal compression,
    - 2-channel version of DXT5A
  - CTX1
    - 2bit selection between 2 8.8bit endpoints
High Dynamic Range Rendering

• Special compact HDR render target format:
  • Just 32 bits: 7e3 7e3 7e3 2
  • Compatible with multisample antialiasing
  • R, G and B are unsigned floating point numbers
    • 7 bits of mantissa
    • 3 bits of exponent
    • Range of 0..16
  • 2 bits of alpha channel
• 16-bit fixed point at half speed
  • With full blending

Displaced subdivision surfaces

Base mesh
• Used by Tessellator to generate vertices

Subdivision surface

Displaced subdivision surface
Questions ?