Xenos: XBOX360 GPU

Michael Doggett
Architect

November 14, 2005

Overview

- Xenos
  - Rendering performance
  - GPU architecture
  - Unified shader
  - Memory Export
  - Texture/Vertex Fetch
  - HDR rendering
  - Displaced subdivision surfaces
- Graphics Hardware
  - GPU Realities
  - Graphics APIs
  - GPU Research

XBOX 360

ATI - Driving the Visual Experience Everywhere

- Products from cell phones to super computers

Gaming

Gaming Console

Integrated

Embedded Display

Notebook

Digital TV

Color Phone Display

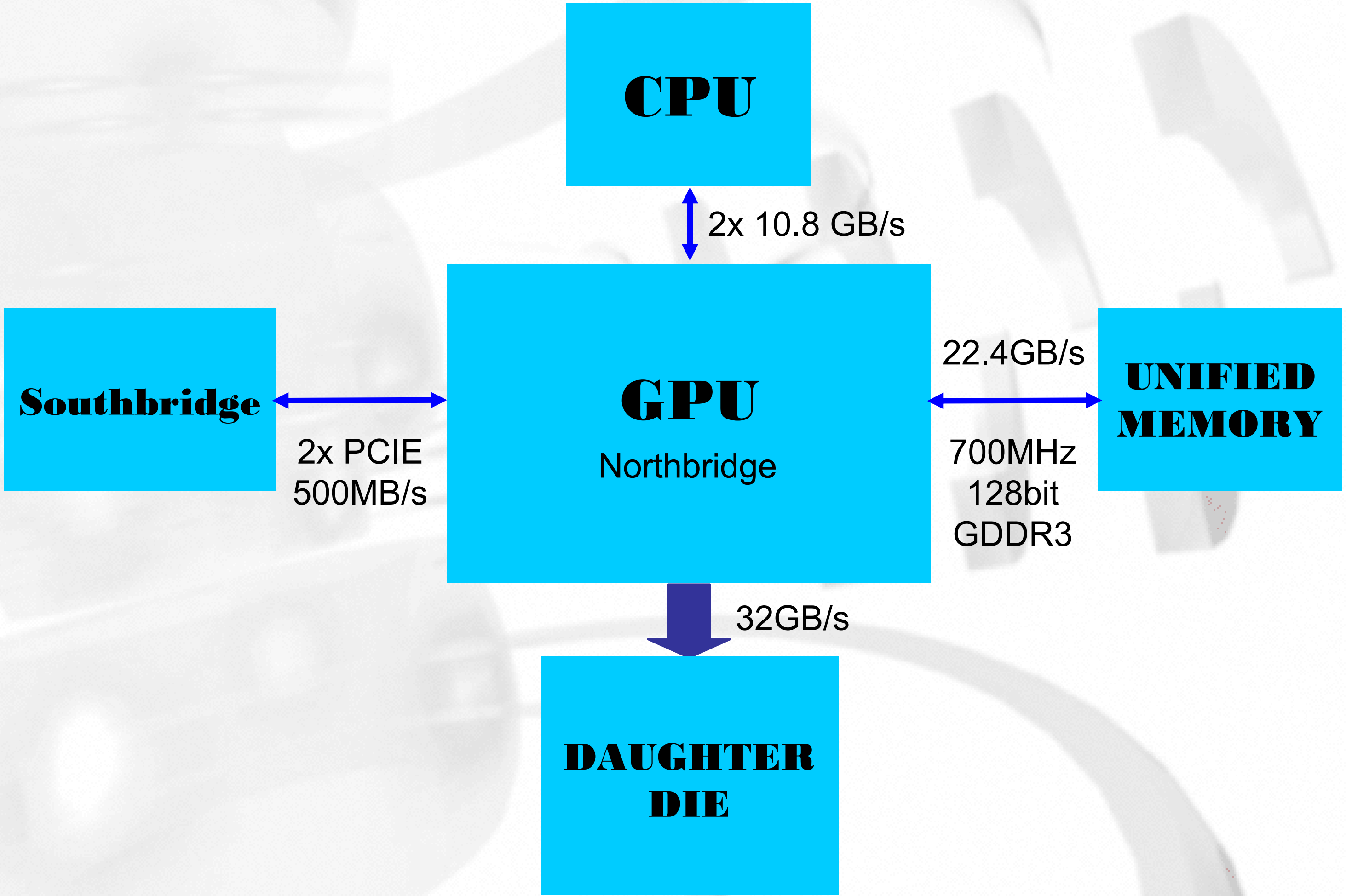
Multimedia

Workstation

Multi Monitor Display

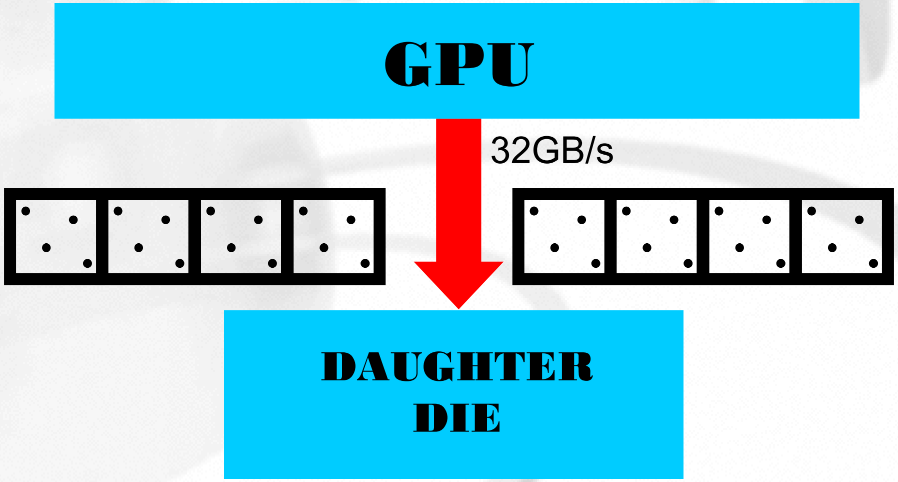
XBOX 360

System architecture



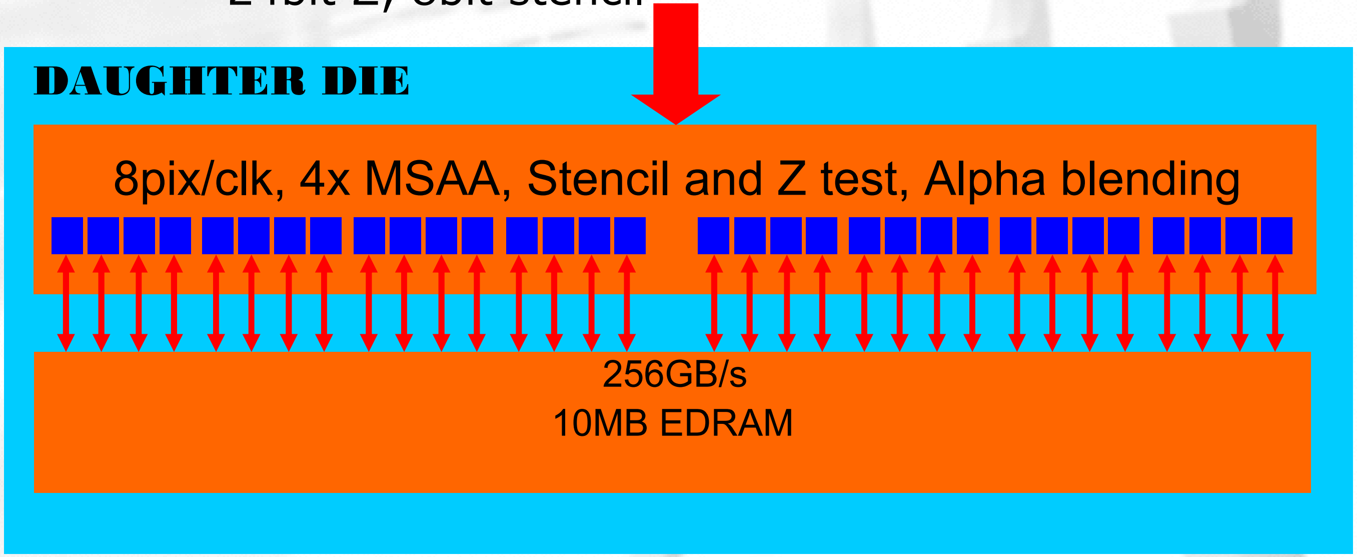
Rendering performance

- 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

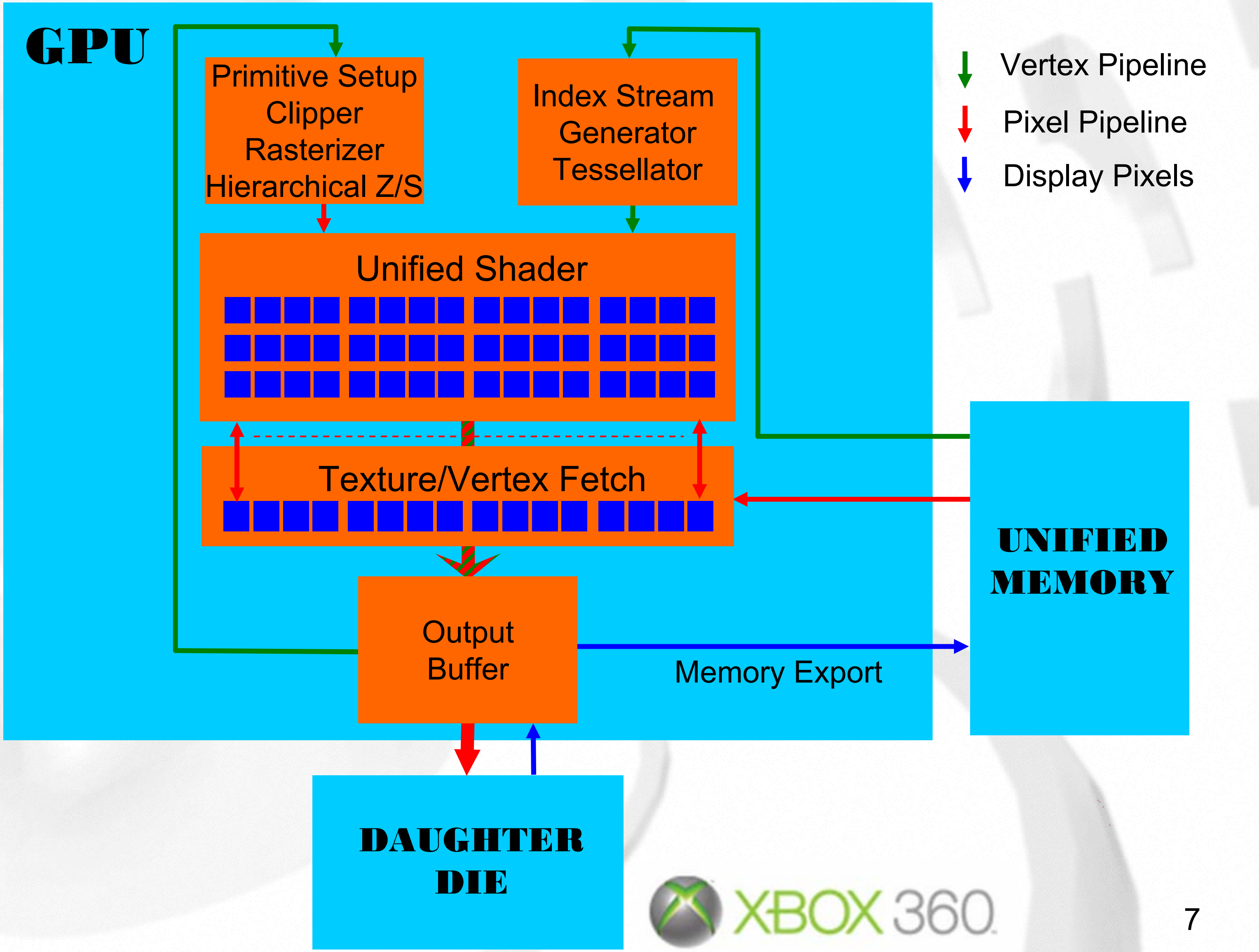


Rendering performance

- 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



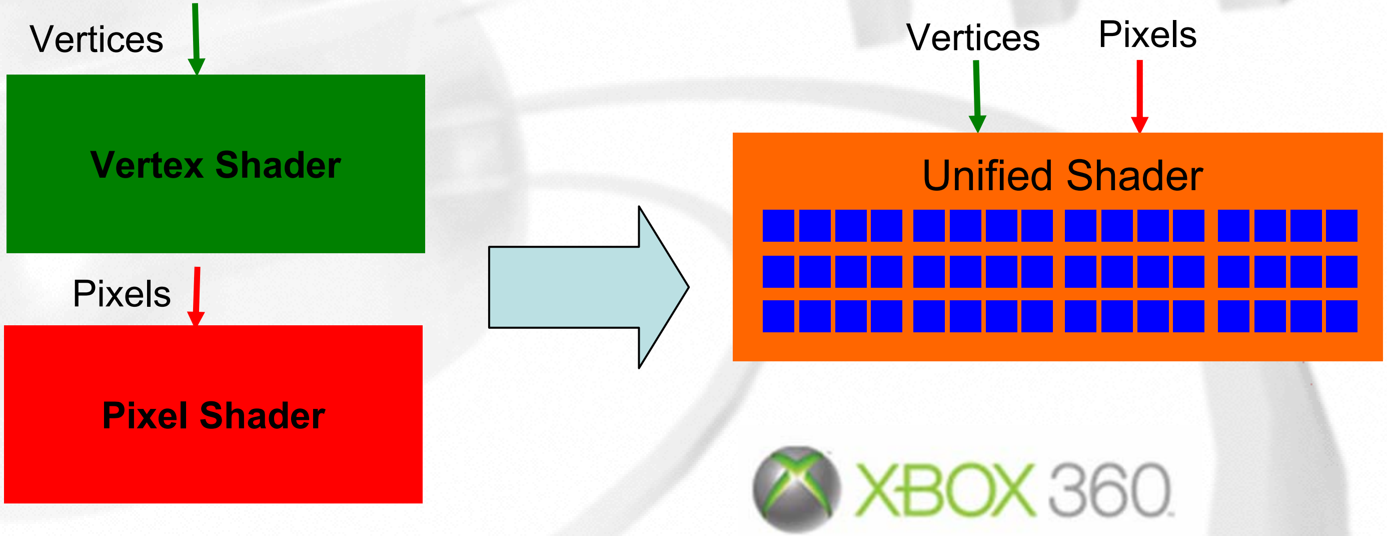
GPU architecture



XBOX 360

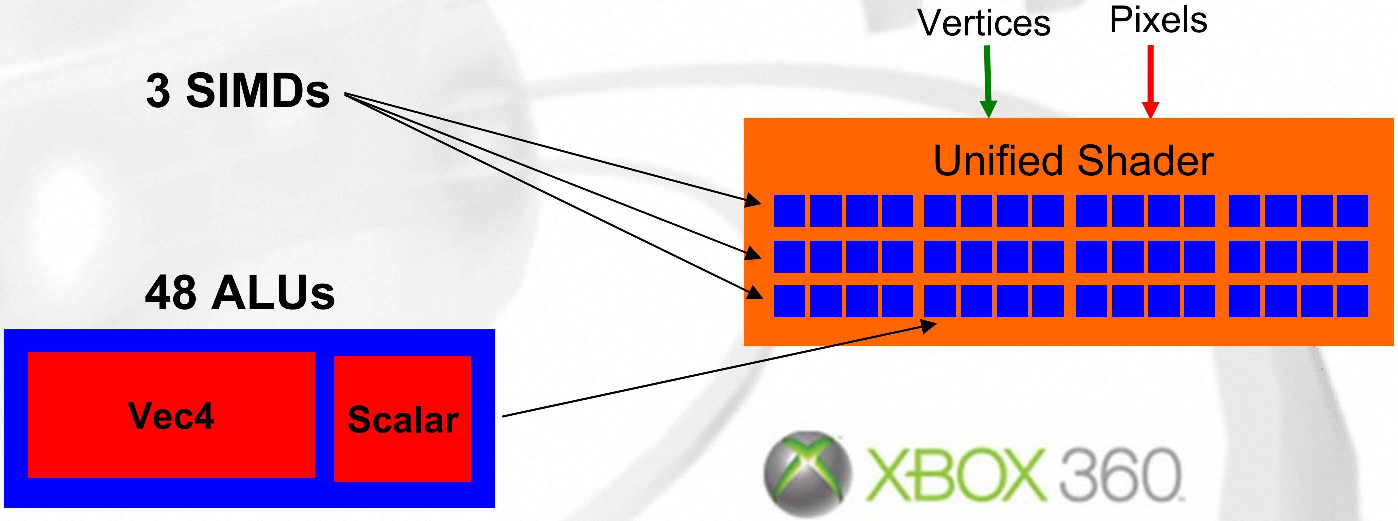
Unified Shader

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



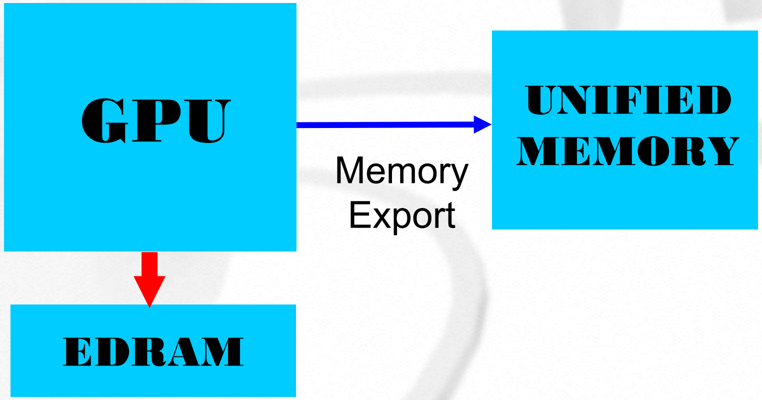
Unified Shader

- 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

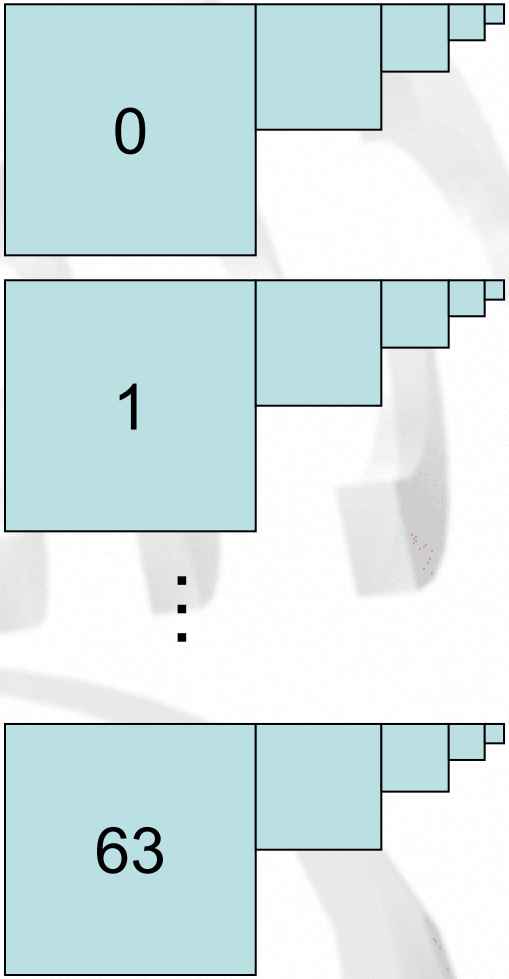


Texture/Vertex Fetch

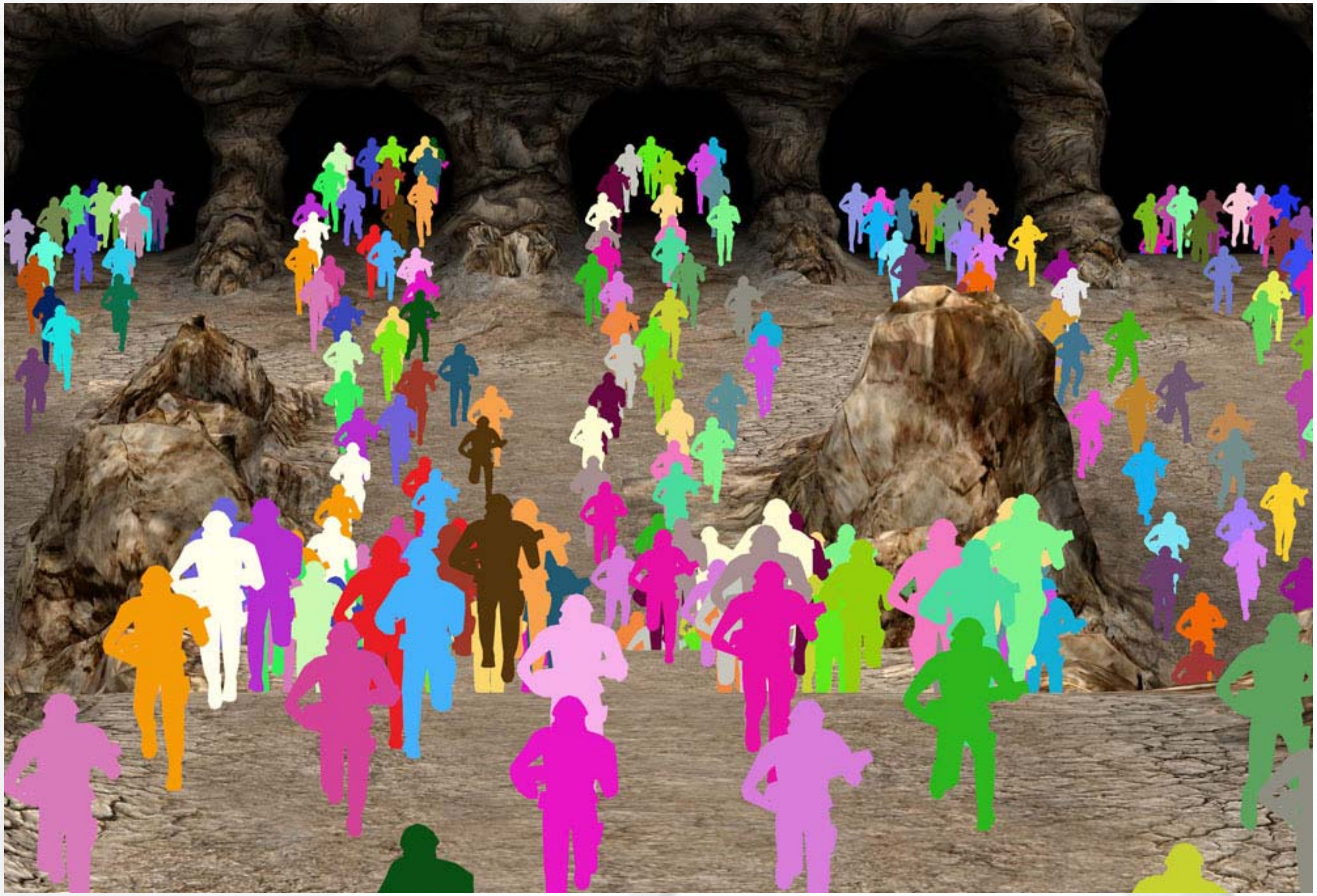
- Shader fetch can be either:
  - Texture fetch (16 units)
    - LOD computation
    - Linear, Bi-linear, Tri-linear Filtering
    - Uses cache optimized for 2D, 3D texture data with varying pixel sizes
    - Unified texture cache
  - Vertex fetch (16 units)
    - Uses cache optimized for vertex-style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

Texture Arrays

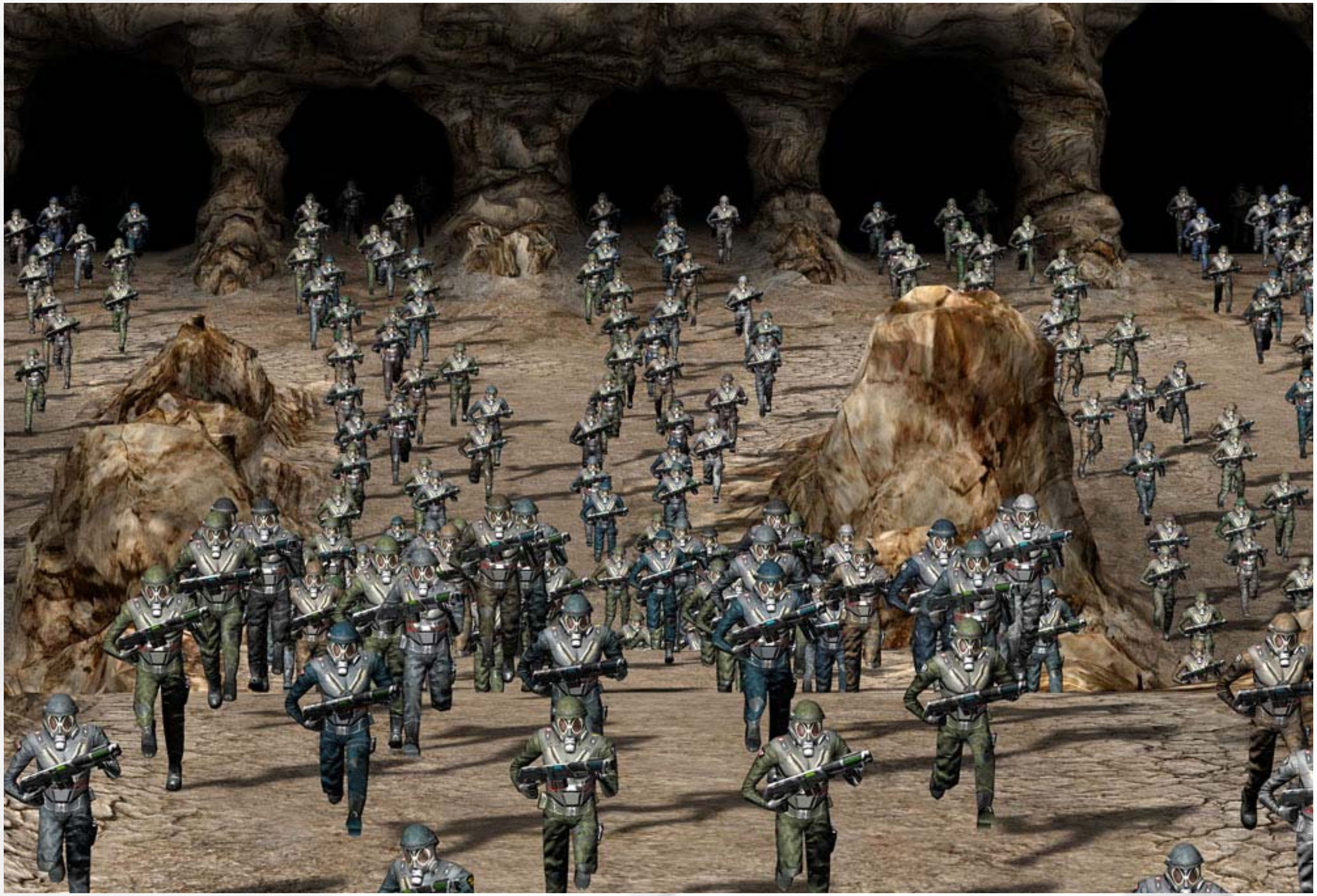
- Generalization of 6 faced cube maps to 64 faces
- Each face is a 2D mip mapped surface
- *Not* volume texture
- Applications
  - Animation frames
  - Varying skins for instanced characters / objects
  - Character shadow texture flipbook animations



Texture array application :
Unique seeds for instanced shading



Texture array application : Hundreds of instanced characters

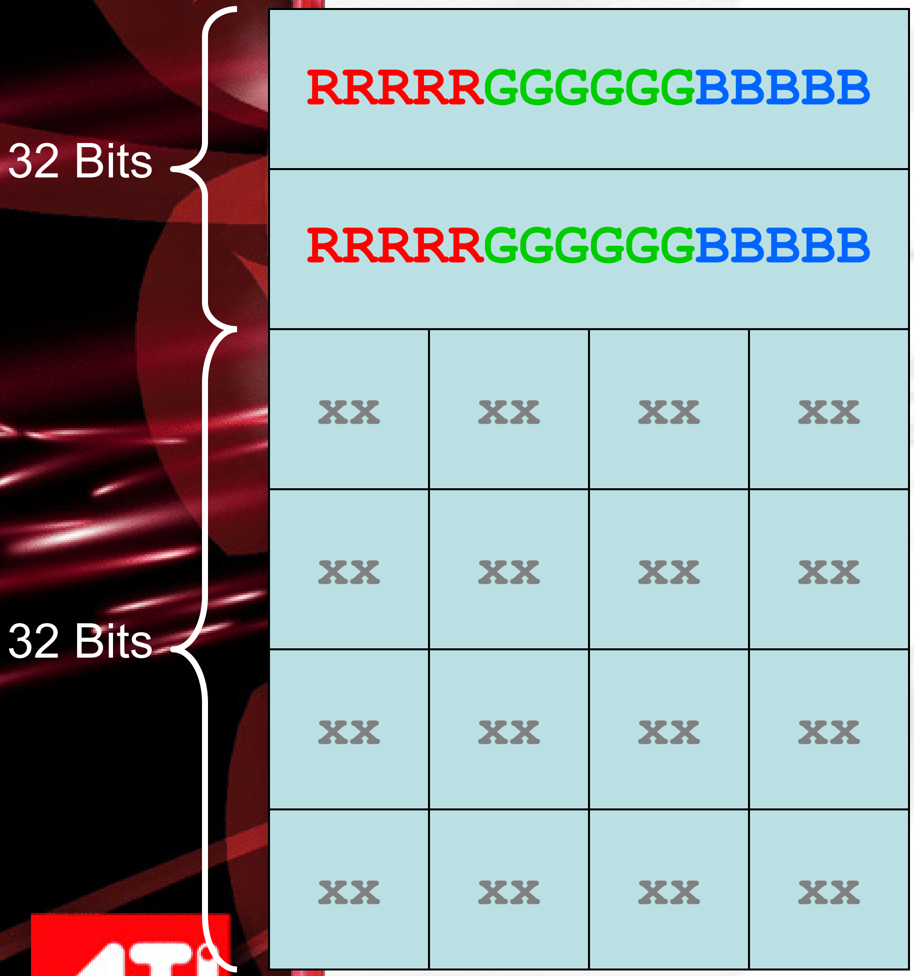


Texture compression

- All of the old DXT formats
  - DXT1, DXT2/3, DXT4/5
- Several new formats (variations on above formats)
  - DXT3A
  - DXT3A as 1111
  - DXT5A
  - DXN
  - CTX1

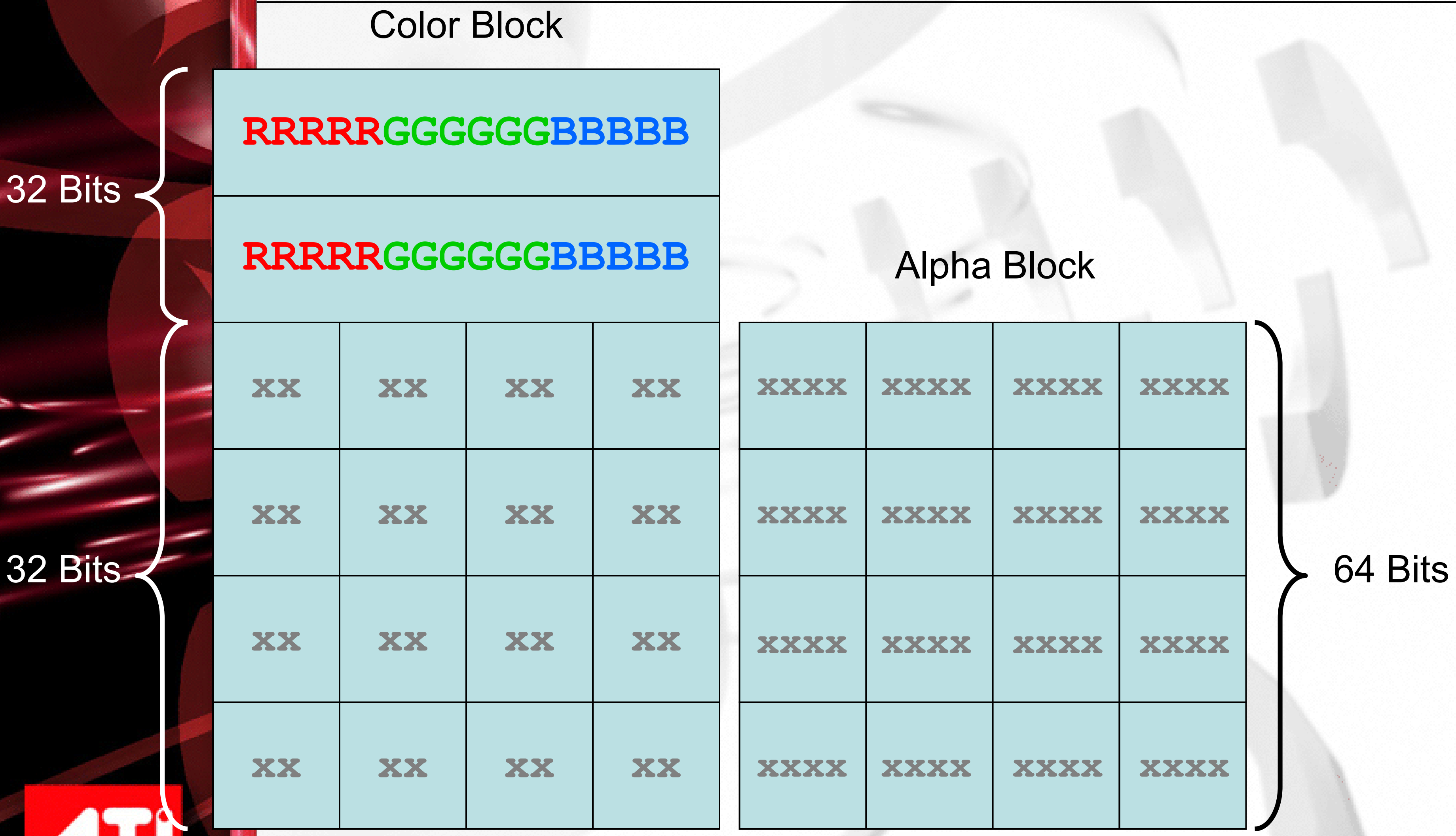
DXT1

Color Block



- Two 565 colors
- Each texel lies on line segment connecting these two colors in RGB space
- Optionally has 1-bit alpha channel depending on relative ordering of RGB colors

DXT2 DXT3



Color Block

32 Bits

RRRRRGGGGGGBBBBB
RRRRRGGGGGGBBBBB

32 Bits

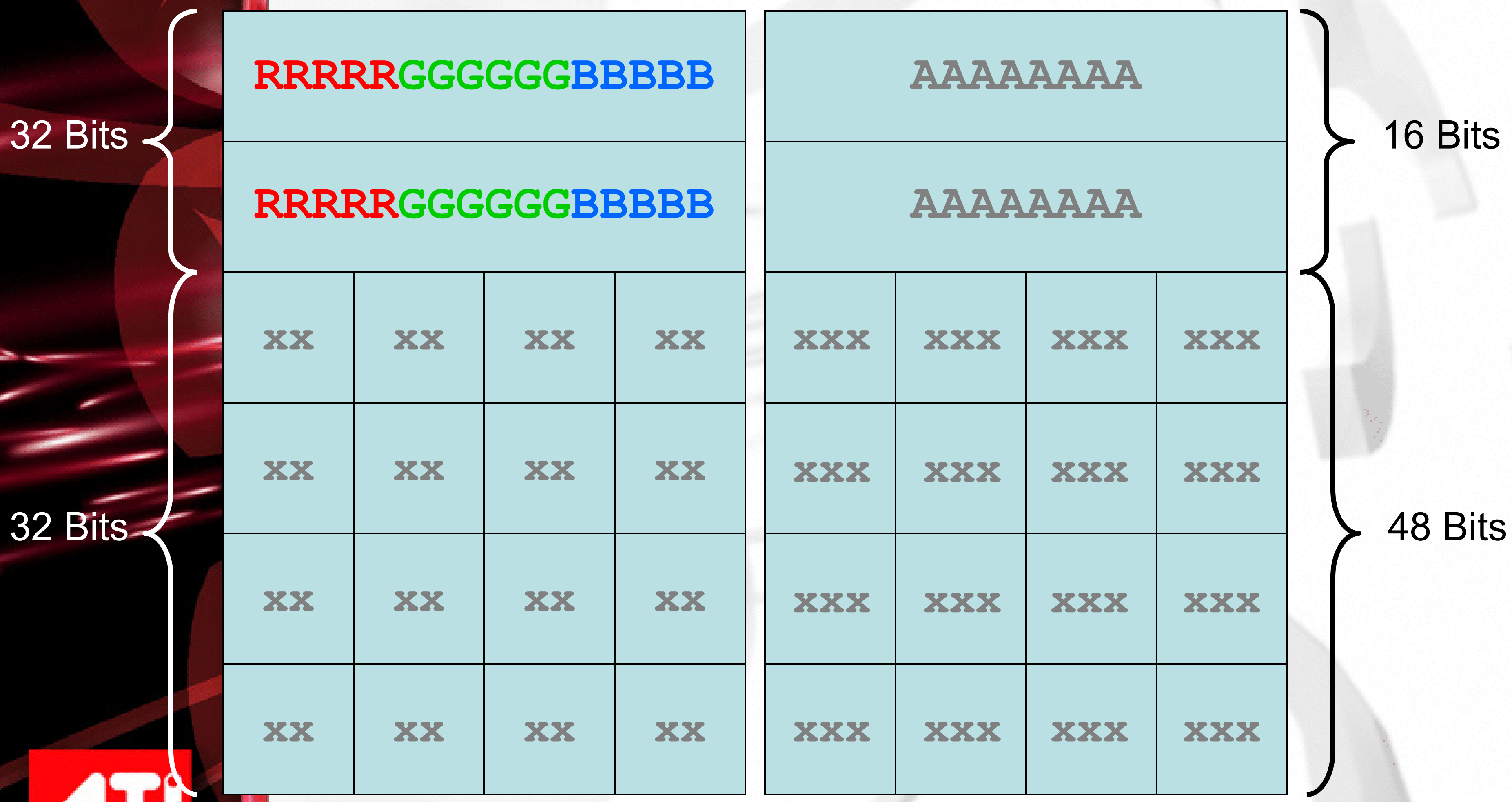
XX	XX	XX	XX
XX	XX	XX	XX
XX	XX	XX	XX
XX	XX	XX	XX

Alpha Block

XXXX	XXXX	XXXX	XXXX
XXXX	XXXX	XXXX	XXXX
XXXX	XXXX	XXXX	XXXX
XXXX	XXXX	XXXX	XXXX

64 Bits

DXT4 DXT5



Color Block

32 Bits

RRRRRGGGGGGBBBBB			
RRRRRGGGGGGBBBBB			
xx	xx	xx	xx
xx	xx	xx	xx
xx	xx	xx	xx
xx	xx	xx	xx

32 Bits

Alpha Block

16 Bits

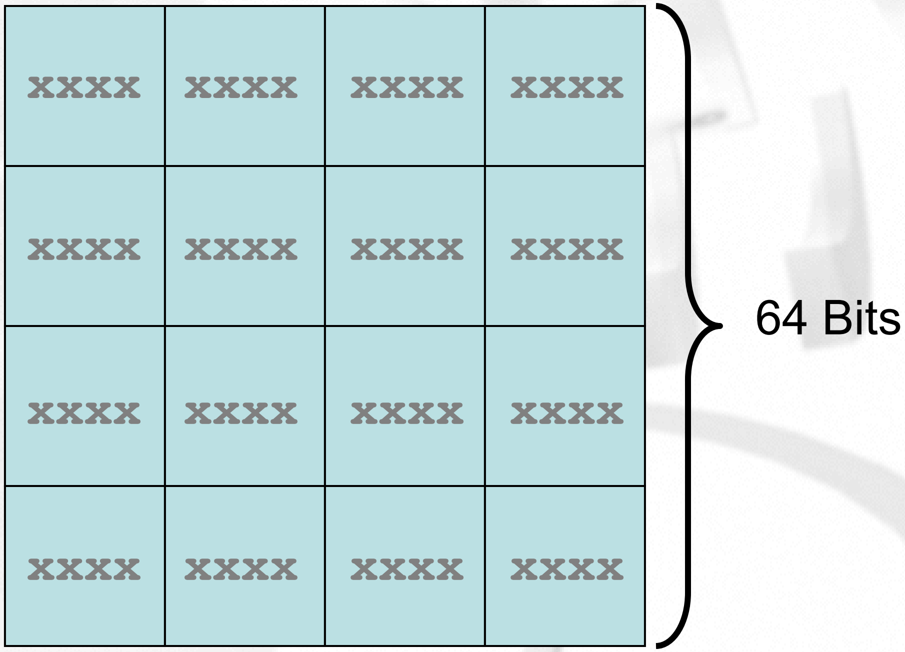
AAAAAAAA			
AAAAAAAA			
xxx	xxx	xxx	xxx
xxx	xxx	xxx	xxx
xxx	xxx	xxx	xxx
xxx	xxx	xxx	xxx

48 Bits

DXT3A

- 4-bit scalar replicated into all channels in pixel shader

Scalar Block

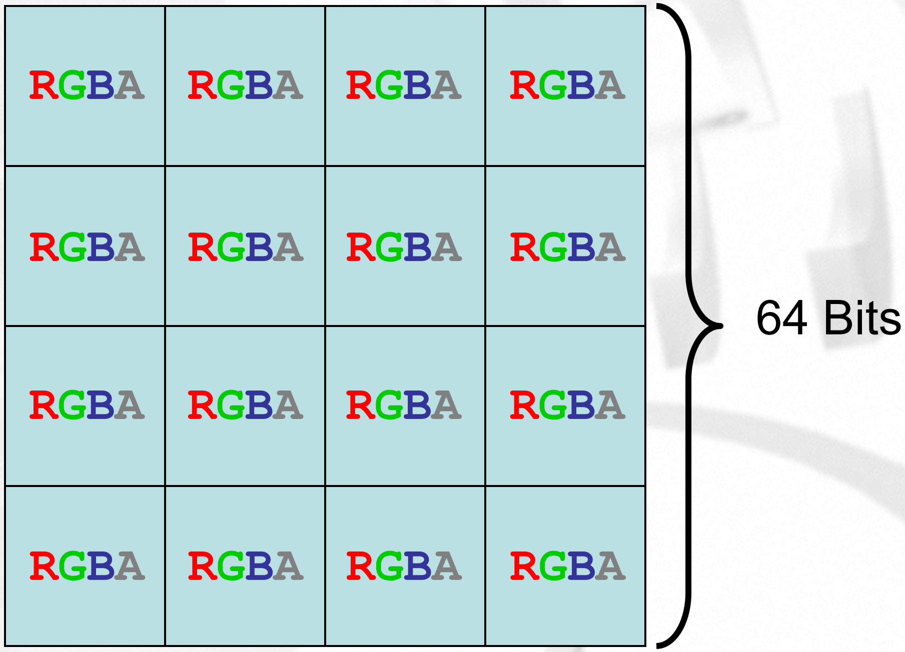


xxxx	xxxx	xxxx	xxxx
xxxx	xxxx	xxxx	xxxx
xxxx	xxxx	xxxx	xxxx
xxxx	xxxx	xxxx	xxxx

64 Bits

DXT3A as 1111

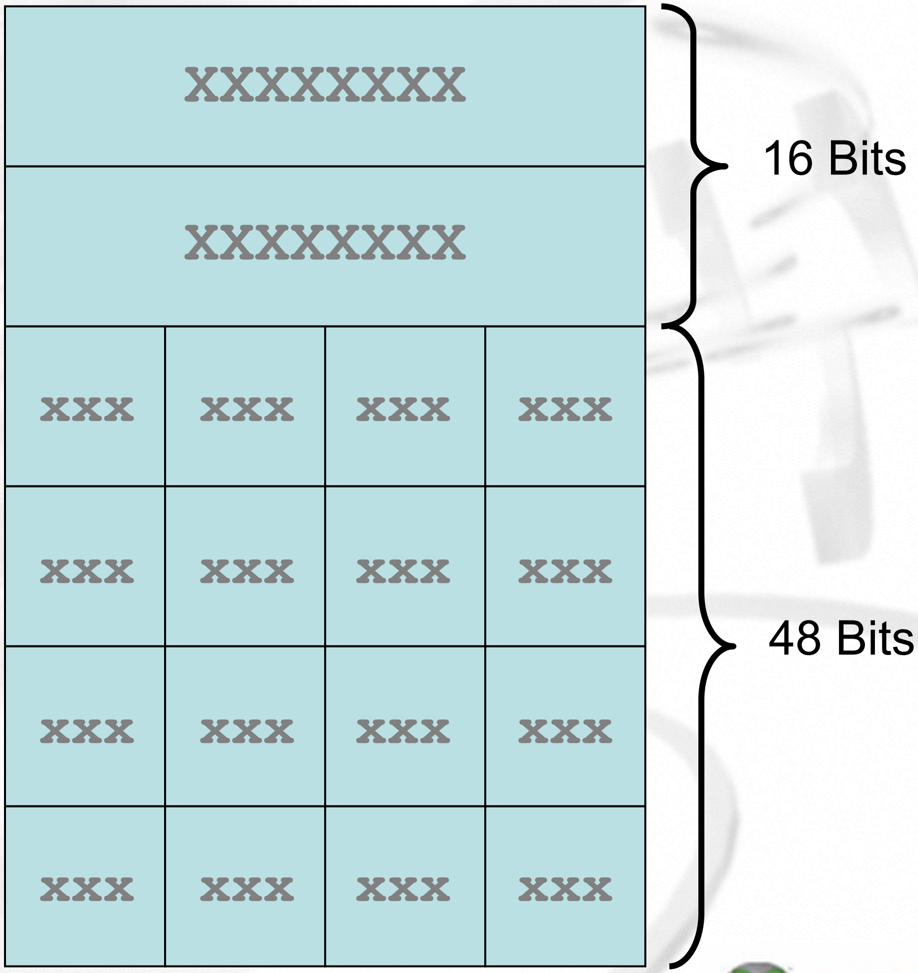
- 1 bit per channel texture
- Filters as you would expect
- Useful for concise masks / logical textures



RGBA	RGBA	RGBA	RGBA
RGBA	RGBA	RGBA	RGBA
RGBA	RGBA	RGBA	RGBA
RGBA	RGBA	RGBA	RGBA

DXT5A

Scalar Block



XXXXXXXX

XXXXXXXX

xxx xxx xxx xxx

xxx xxx xxx xxx

xxx xxx xxx xxx

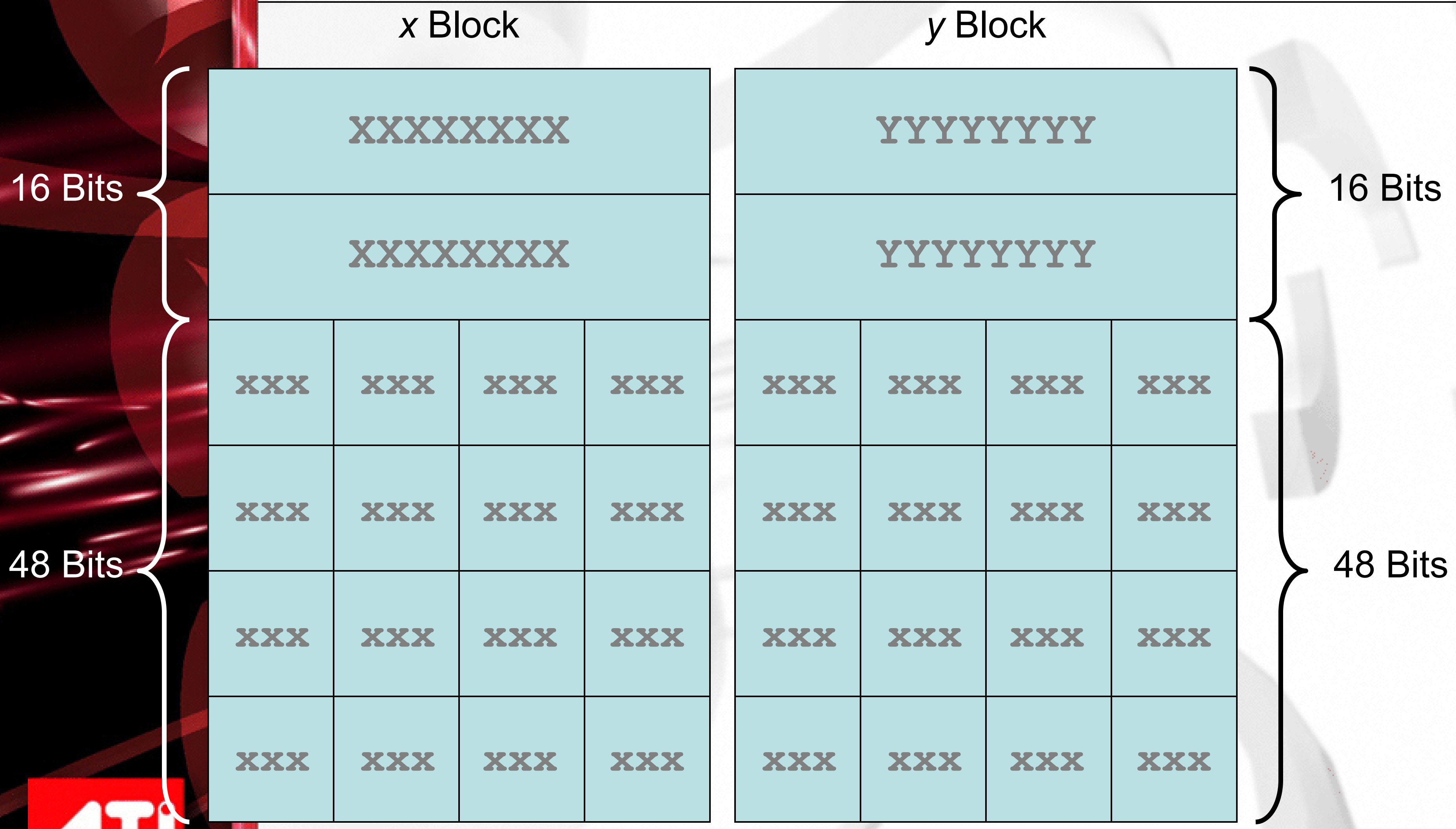
xxx xxx xxx xxx

16 Bits

48 Bits

DXN

- Same as 3Dc normal compression
- Two-channel version of DXT5A
- Good for tangent-space normal maps



x Block

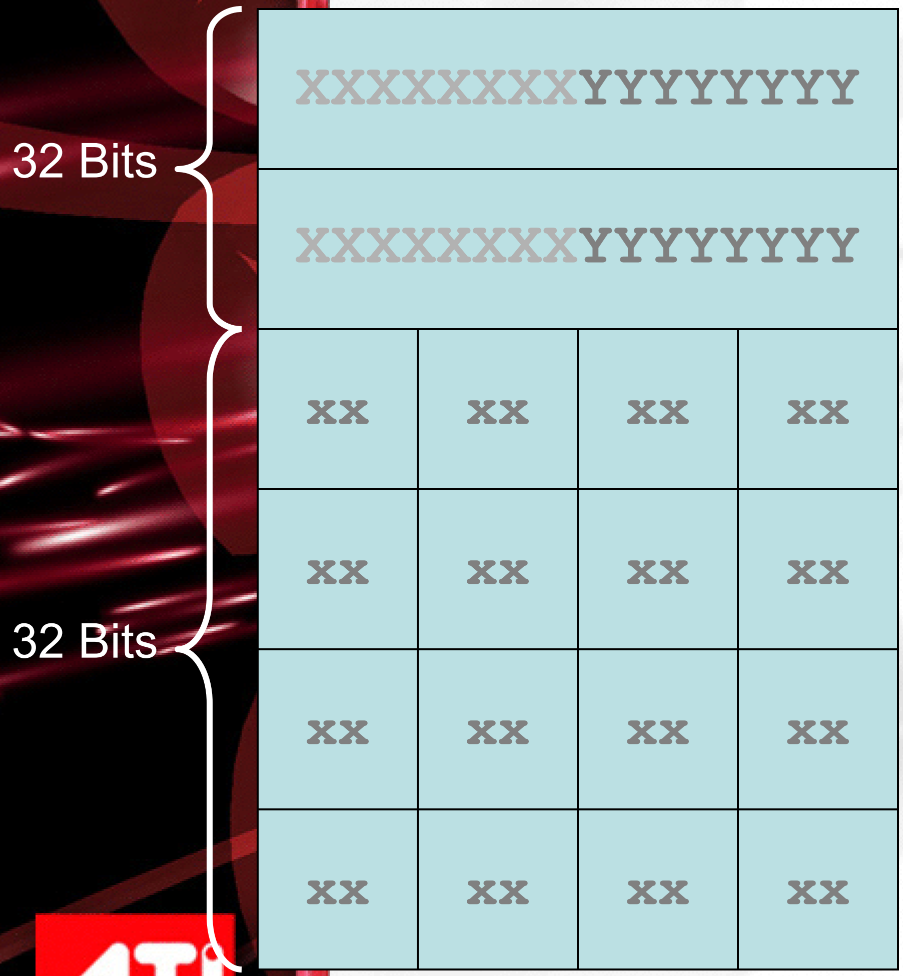
XXXXXXXX			
XXXXXXXX			
xxx	xxx	xxx	xxx
xxx	xxx	xxx	xxx
xxx	xxx	xxx	xxx
xxx	xxx	xxx	xxx

y Block

YYYYYYYY			
YYYYYYYY			
xxx	xxx	xxx	xxx
xxx	xxx	xxx	xxx
xxx	xxx	xxx	xxx
xxx	xxx	xxx	xxx

CTX1

XY Block



32 Bits

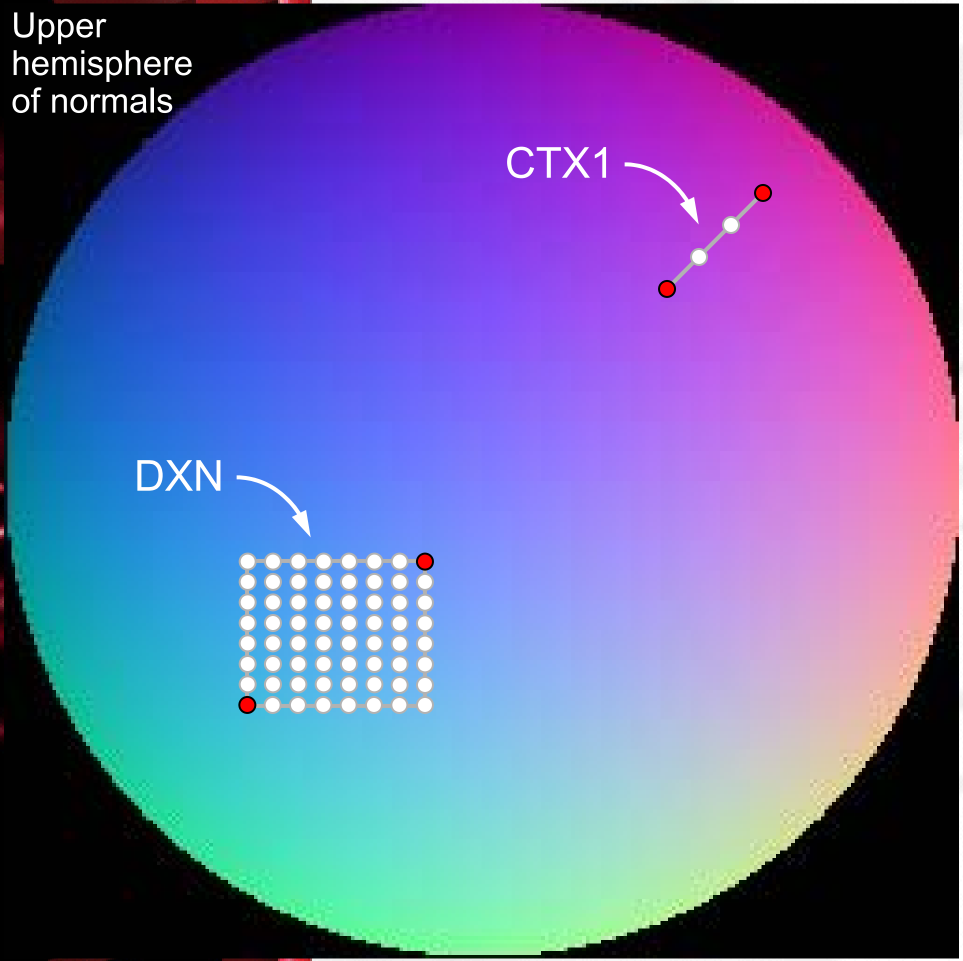
XXXXXXXXYYYYYYYY			
XXXXXXXXYYYYYYYY			

32 Bits

xx	xx	xx	xx
xx	xx	xx	xx
xx	xx	xx	xx
xx	xx	xx	xx

- Two-channel format with channels sharing two-bit interpolation value
- Lower quality but more concise tangent-space normal maps than DXN

CTX1 vs. DXN



- Both have 8.8 anchor points
- CTX1
  - *x* and *y* share 2-bit interpolation value
  - 4 representable normals per 4×4 block of texels
  - 4 bits per texel
- DXN
  - Independent 3-bit interpolation values for *x* and *y*
  - 64 representable normals per 4×4 block of texels
  - 8 bits per texel

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

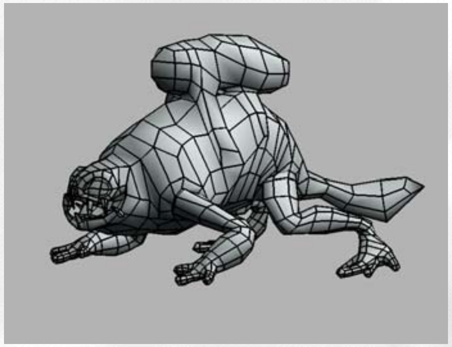
Displaced subdivision surfaces

- Prototype algorithm
- Vineet Goel, ATI research Orlando

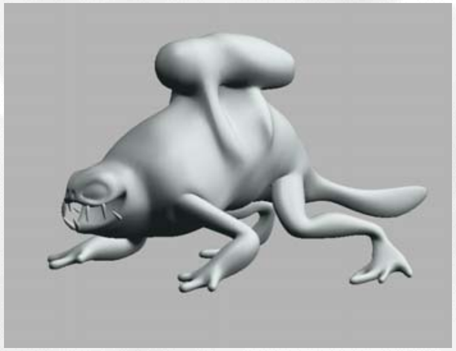
Displaced subdivision surface algorithm

- Tessellator:
  - Generates 64 vertices for each patch that are fed into the VS.
- Vertex Shader:
  - Reads in one-ring, computes Stam’s method using precomputed table lookup
  - Adds Displacement map
- Pixel Shader
  - Adds bump mapping and surface color

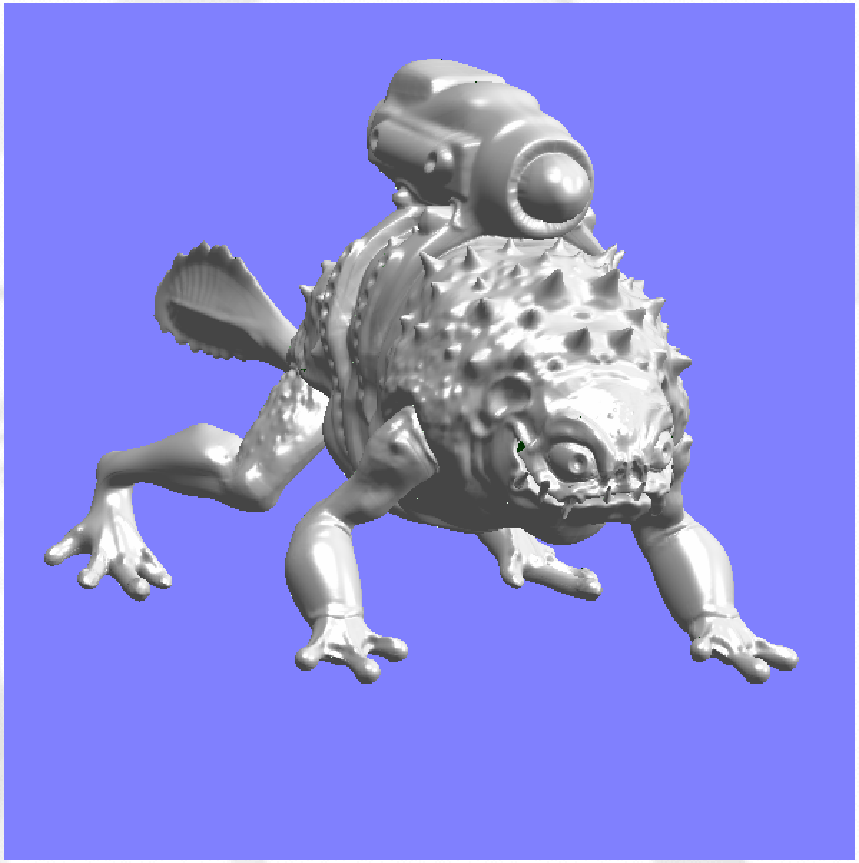
Displaced subdivision surfaces



Base mesh
- Used by Tessellator to generate vertices



Subdivision surface



Displaced subdivision surface

Graphics Hardware

GPU realities

- Die size/cost/yield
- One die, multiple products
- Decreasing technology, increasing power
- How to cool large chips ?
- Ensure design is scalable
- Refinements and enhancements of the current pipeline

Graphics APIs

- Windows Vista
  - Virtual Memory
  - Improved state change efficiency
- Vista DirectX9
  - ClearType
- DirectX10
  - Unified shader programming model
  - Geometry Shader
    - Access to entire triangle and adjacent vertices
    - Output to array of render targets, cube maps
  - Stream output from Geometry Shader

GPU research

- Improved multi-GPU performance and antialiasing
  - CrossFire
  - Multi-chip, multi-core
- Higher Order Surfaces
  - Subdivision surfaces, NURBS
- General Purpose GPU (GPGPU)
  - Driving improved 32bit float performance
  - Xenos shader scatter write

GPU research - Shadows

- Performance enhancements for stencil shadows and shadow buffers
- Hierarchical rendering
  - User low resolution shadow map to find areas that require detail rasterization
    - Chan, Durand EGSR04
  - Stencil shadow volumes using 8x8 pixel tiles
    - Aila, Akenine-Möller GH04
- Improving Z, stencil performance

GPU research – Ray Tracing

- Accelerating ray tracing on GPUs
  - Build and update acceleration structures
    - Xenos shader scatter write
- GPU evolution affected by ray tracing

Conclusion

- Xenos
  - Architecture
  - Features
- Graphics Hardware
  - Future
  - Research

Questions ?