NV1
SEGA Virtua Fighter
50K polygons/sec
1M pixel ops/sec
Circa 1995
Convergence of Film and Real-time Rendering
Cinematic Shading

Final Fantasy
The Spirits Within
Square
Real-time Cinematic Shading requires new levels of features and performance

- Advanced Programmability
- High-precision color
- High-level Shading Language
- Highly efficient architecture
- High bandwidth to system memory and CPU

Artist: Count Love
Introducing the CineFX Architecture

- Generalized Vertex Processing
- Generalized Pixel Processing
- 128-bit Floating Point Precision
- Highly advanced rendering architecture
- Dramatically improved performance
CineFX
Generalized Vertex Processing

- Up to 65536 vertex instructions
- 256 constants
- Loops & Branching
  - Forward & backwards
  - Data Dependent
- Call & Return - Subroutines
- Per component condition codes & write masks
  - Faster than branching for short basic blocks

<table>
<thead>
<tr>
<th></th>
<th>DX8.0</th>
<th>R300</th>
<th>CineFX</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vertex Shaders</td>
<td>1.1</td>
<td>2.0</td>
<td>2.0+</td>
</tr>
<tr>
<td>Max Instructions</td>
<td>128</td>
<td>1024</td>
<td>65536</td>
</tr>
<tr>
<td>Max Static Instructions</td>
<td>128</td>
<td>256</td>
<td>256</td>
</tr>
<tr>
<td>Max. Constants</td>
<td>96</td>
<td>256</td>
<td>256</td>
</tr>
<tr>
<td>Temporary Registers</td>
<td>12</td>
<td>12</td>
<td>16</td>
</tr>
<tr>
<td>Max Loops</td>
<td>0</td>
<td>4</td>
<td>256</td>
</tr>
<tr>
<td>Conditional Write Masks</td>
<td>-</td>
<td>-</td>
<td>✓</td>
</tr>
<tr>
<td>Call &amp; Return</td>
<td>-</td>
<td>-</td>
<td>✓</td>
</tr>
<tr>
<td>Static Flow Control</td>
<td>-</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Dynamic Flow Control</td>
<td>-</td>
<td>-</td>
<td>✓</td>
</tr>
</tbody>
</table>
CineFX

Vertex Processing Instruction Set

- **Add & multiply instructions**
  ADD, DP3, DP4, DPH, MAD, MOV, SUB

- **Math functions**
  ABS, COS, EX2, EXP, FLR, FRC, LG2, LOG, RCP, RSQ, SIN

- **Set on instructions**
  SEQ, SFL, SGR, SGT, SLE, SLT, SNE, STR

- **Branching instructions**
  BRA, CAL, RET

- **Address register instructions**
  ARL, ARA

- **Graphics-oriented instructions**
  DST, LIT, RCC, SSG

- **Minimum / maximum instructions**
  MAX, MIN
CineFX

Vertex Processing Example – Matrix Palette Skinning

DX8 / NV2x / R200 / RV250
- 4 shaders
  - 1 bone
  - 2 bone
  - 3 bone
  - 4 bone
- Segment Model into those polys depending on 1,2,3,4 bones
- Draw separately

DX9 / R300
- 1 shader (1-4 bones)
- Branching is *per-object*
- Still have to segment model into 1-4 bone groups
- Draw separately

CineFX
- 1 shader
- Branching is *per-vertex*
- No need to segment model
- Loop is done conditionally on a per-vertex level
CineFX

*Generalized Pixel Shading*

- All the features of GeForce4 vertex programs – for pixels!
- Full instruction set for pixels
- Up to 1024 shader instructions
- Up to 16 textures per pixel
- Instruction predicates for conditional execution
- Per component swizzling
- Per component write masks
- Arbitrary Texture Filters

<table>
<thead>
<tr>
<th></th>
<th>DX8.0</th>
<th>R300</th>
<th>NV30</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Pixel Shaders</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Texture Maps</td>
<td>1.1</td>
<td>2.0</td>
<td>2.0+</td>
</tr>
<tr>
<td>Max. Texture Instructions</td>
<td>4</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>Max. Color Instructions</td>
<td>4</td>
<td>32</td>
<td>1024</td>
</tr>
<tr>
<td>Max Temp Storage</td>
<td>8</td>
<td>64</td>
<td>1024</td>
</tr>
<tr>
<td>Instruction Predicates</td>
<td>-</td>
<td>-</td>
<td>✓</td>
</tr>
<tr>
<td>Unlimited Dependent Textures</td>
<td>-</td>
<td>-</td>
<td>✓</td>
</tr>
<tr>
<td>Swizzling</td>
<td>-</td>
<td>-</td>
<td>✓</td>
</tr>
<tr>
<td>Advanced Instructions</td>
<td>-</td>
<td>-</td>
<td>✓</td>
</tr>
<tr>
<td>Conditional Write Masks</td>
<td>-</td>
<td>-</td>
<td>✓</td>
</tr>
</tbody>
</table>
CineFX

*Pixel Shading Instruction Set*

- **Add & multiply instructions**
  - ADD, DP3, DP4, LRP, MAD, MOV, SUB, X2D

- **Texturing instructions**
  - TEX, TXD, TXP

- **Partial derivative instructions**
  - DDX, DDY

- **Math functions**
  - COS, EX2, FLR, FRC, LG2, POW, RCP, RSQ, SIN

- **Set on instructions**
  - SEQ, SFL, SGR, SGT, SLE, SLT, SNE, STR

- **Graphics-oriented instructions**
  - DST, LIT, RFL

- **Minimum / maximum instructions**
  - MAX, MIN

- **Pack instructions**
  - PK2H, PK2US, PK4B, PK4UB, PK4UBG

- **Unpack instructions**
  - UP2H, UP2US, UP4B, UP4UB, UP4UBG

- **Kill instruction**
  - KIL
CineFX
Unprecedented Precision

Record-breaking 128-bit color precision
- Support for 16 or 32-bit floating point components
- 64-bit & 128-bit FP color
- 64-bit offers improved precision with 2x the performance & ½ the memory of 128-bit

<table>
<thead>
<tr>
<th></th>
<th>DX8</th>
<th>R300</th>
<th>CineFX</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max Precision</td>
<td>32</td>
<td>96</td>
<td>128</td>
</tr>
<tr>
<td>32-bit color</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>64-bit color</td>
<td>-</td>
<td>-</td>
<td>✓</td>
</tr>
<tr>
<td>128-bit color</td>
<td>-</td>
<td>-</td>
<td>✓</td>
</tr>
<tr>
<td>Format</td>
<td>INT</td>
<td>FP</td>
<td>FP</td>
</tr>
</tbody>
</table>
High Precision Color Details

- 16-bit floating point format
  - The same exact format that Pixar & ILM use for films
  - So-called s10e5 representation
    - 1 bit sign
    - 10 bit mantissa
    - 5 bit exponent, -15 bias
  - Otherwise IEEE 754 floating-point semantics
  - More range than signed shorts
  - Half the space & bandwidth of a 32-bit floating-point value

- 32-bit Floating Point format: IEEE s23e8
CineFX
High Dynamic Range

Overdark - Underexposed  |  Standard Exposure  |  Overbright - Overexposed
CineFX
Advanced Graphics Processing

- Render to Vertex Array
  - Displacement Mapping
  - Particle Systems
- Ray Tracing
- Floating Point Cinematic precision compositing
Convergence of Film and Real-time Rendering

A Step Function Discontinuity Toward Cinematic Rendering

FILM

Real-time

NVIDIA NV30 + Cg

NVIDIA CONFIDENTIAL
Cg
What developers have been asking for

C for Graphics – Cg Shader Code written in a High Level Language
Compiled and optimized
Into Vertex and Pixel Shader Assembly Code
100% compatible with DX9’s HLSL

“Cg is a bright light for a black art. Finally cryptic shader tricks can be explored and applied by mere mortals.”

Mike Biddlecombe
Programmer - Dungeon Siege
Gas Powered Games
NVIDIA Cg Compiler – Breakthrough Technology

- Optimized
  - Generates the fastest code for NVIDIA GPUs

- Flexible
  - Outputs DirectX or OpenGL shader programs
  - Supports Windows, Linux, Mac OS X, Xbox

- Compatible
  - Forwards to: DirectX 9, NV30...
  - Backwards to: ALL DirectX 8 / OpenGL 1.4 compliant GPUs
Key DCC Applications integrating Cg

3ds max™

Maya®

Softimage|XSI™

NVIDIA CONFIDENTIAL
CineFX Architecture

- Generalized Vertex Processing
- Generalized Pixel Processing
- 128-bit Floating Point Precision
- Highly advanced rendering architecture
- Dramatically improved performance