NEW GPU FEATURES
OF NVIDIA’S MAXWELL ARCHITECTURE

ALEXEY PANTELEEV
DEVELOPER TECHNOLOGY ENGINEER, NVIDIA
OUTLINE

- Architectural goals of Maxwell
- DirectX12 hardware features
  - Conservative Rasterization
  - Raster Order Views
  - Tiled Resources
- Multi-Projection Acceleration
- New Antialiasing Features
- Misc other new features
- Questions and Answers
MAXWELL ARCHITECTURAL GOALS

- New architecture for improved efficiency
- Massively improved perf / watt
  - Still on a 28nm process
- Focus on new graphics features
  - Real-time GI for rich dynamic scenes
  - Higher quality, programmable AA
  - Working set management
  - SVG rendering acceleration
  - Create the best platform for DirectX 12
MAXWELL ARCHITECTURAL GOALS

- New architecture for improved efficiency
- Massively improved perf / watt
  - Still on a 28nm process
- Focus on new graphics features
  - Real-time GI for rich dynamic scenes
  - Higher quality, programmable AA
  - Working set management
  - SVG rendering acceleration
  - Create the best platform for DirectX 12

<table>
<thead>
<tr>
<th></th>
<th>680</th>
<th>780</th>
<th>980</th>
</tr>
</thead>
<tbody>
<tr>
<td>TFLOPS</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>MEMORY</td>
<td>2GB</td>
<td>3GB</td>
<td>4GB</td>
</tr>
<tr>
<td>PERFORMANCE</td>
<td>1</td>
<td>1.5</td>
<td>2</td>
</tr>
<tr>
<td>POWER</td>
<td>195W</td>
<td>250W</td>
<td>165W</td>
</tr>
<tr>
<td>GFLOPS / WATT</td>
<td>15</td>
<td>15</td>
<td>30</td>
</tr>
</tbody>
</table>
New architecture for improved efficiency

- Massively improved perf / watt
  - Still on a 28nm process

Focus on new graphics features

- Real-time GI for rich dynamic scenes
- Higher quality, programmable AA
- Working set management
- SVG rendering acceleration
- Create the best platform for DirectX 12
MAXWELL ARCHITECTURAL GOALS

- New architecture for improved efficiency
- Massively improved perf / watt
  - Still on a 28nm process
- Focus on new graphics features
  - Real-time GI for rich dynamic scenes
  - Higher quality, programmable AA
  - Working set management
  - SVG rendering acceleration
  - Create the best platform for DirectX 12
MAXWELL ARCHITECTURAL GOALS

- New architecture for improved efficiency
- Massively improved perf / watt
  - Still on a 28nm process
- Focus on new graphics features
  - Real-time GI for rich dynamic scenes
  - Higher quality, programmable AA
  - Working set management
  - SVG rendering acceleration
  - Create the best platform for DirectX 12
MAXWELL ARCHITECTURAL GOALS

- New architecture for improved efficiency
- Massively improved perf / watt
  - Still on a 28nm process
- Focus on new graphics features
  - Real-time GI for rich dynamic scenes
  - Higher quality, programmable AA
  - Working set management
  - SVG rendering acceleration
  - Create the best platform for DirectX 12
MAXWELL ARCHITECTURAL GOALS

- New architecture for improved efficiency
- Massively improved perf / watt
  - Still on a 28nm process
- Focus on new graphics features
  - Real-time GI for rich dynamic scenes
  - Higher quality, programmable AA
  - Working set management
  - SVG rendering acceleration
  - Create the best platform for DirectX 12
DIRECTX 12 FEATURES

- New API is parallelizable for rendering on multicore CPUs
- Reduced API overhead for single-core work
- More nimble resource binding model using indexing
- More efficient data management/transfer model
- More explicit work scheduling model
- New hardware features
OUTLINE

- Architectural goals of Maxwell
- DirectX12 hardware features
  - Conservative Rasterization
  - Raster Order Views
  - Tiled Resources
- Multi-Projection Acceleration
- New Antialiasing Features
- Misc other new features
- Questions and Answers
REGULAR RASTERIZATION

- Test each pixel center
- Include fragments with center covered
- Small triangles can be dropped

- Can’t easily create data structures
  - E.g. triangle lists for ray tracing
CONSERVATIVE RASTERIZATION

- Draws all pixels a triangle touches
  - Different Tiers - see DX spec

- Possible before through GS trick but relatively slow
  - See J. Hasselgren et al. “Conservative Rasterization“, GPU Gems 2

- Now we can use rasterization to implement some nice techniques!
HYBRID RAYTRACED SHADOWS


J. Story “Hybrid Ray-Traced Shadows“, D3D Day GDC 2015

- Rasterize light view conservatively
- Store triangle info in buffers:
  - Vertex Buffer
  - $NxNxd$ Prim Indices Map
  - $NxN$ Prim Count Map
- Raytrace triangles in a later pass
RAYTRACED SHADOWS DEMO
OUTLINE

- Architectural goals of Maxwell
- DirectX12 hardware features
  - Conservative Rasterization
  - Raster Order Views
  - Tiled Resources
- Multi-Projection Acceleration
- New Antialiasing Features
- Misc other new features
- Questions and Answers
Pixel shader writes to UAVs are unordered
  - Can’t guarantee determinism

Can’t do...
  - Programmable Blending
  - Smart OIT implementations
  - Arbitrary g-buffer data packing
  - Other per-pixel data structures
RASTER ORDER VIEWS (ROV)

- ROVs guarantee ordering and atomicity

- Ordering doesn’t come for free
  - Depth complexity affects performance

- Always compare with other options
  - Advanced blending operations
  - Atomics, lock-free algorithms
OUTLINE

- Architectural goals of Maxwell
- DirectX12 hardware features
  - Conservative Rasterization
  - Raster Order Views
  - Tiled Resources
- Multi-Projection Acceleration
- New Antialiasing Features
- Misc other new features
- Questions and Answers
DX12 TILED RESOURCES

- Full support for tiled 3D Textures/Arrays
  - On top of what DX11.2 provides

- Enable fine grained working set management
- Texture defined as a set of 64 KB tiles
- Memory for tiles is allocated separately
TILED RESOURCES APPLICATIONS

- Fine-grained working set management
  - Texture streaming, Clip-maps

- Variable resolution resources
  - Adaptive shadow maps
  - Sparse multi-resolution rendering

- Sparse representation
  - Voxel grids
  - Simulation - physics, path finding
TILED RESOURCES APPLICATIONS

- Fine-grained working set management
  - Texture streaming, Clip-maps

- Variable resolution resources
  - Adaptive shadow maps
  - Sparse multi-resolution rendering

- Sparse representation
  - Voxel grids
  - Simulation - physics, path finding
TILED RESOURCES APPLICATIONS

- Fine-grained working set management
  - Texture streaming, Clip-maps

- Variable resolution resources
  - Adaptive shadow maps
  - Sparse multi-resolution rendering

- Sparse representation
  - Voxel grids
  - Simulation - physics, path finding
SPARSE FLUID SIMULATION

- Uses tiled resources to only simulate/store grid cells that contain fluid
- Save computation time and memory
- See Alex Dunn, ”Sparse Fluid Simulation in DirectX” at GTC’15 Thursday 2:30 PM
SPARSE FLUID DEMO
OUTLINE

- Architectural goals of Maxwell
- DirectX12 hardware features
  - Conservative Rasterization
  - Raster Order Views
  - Tiled Resources
- Multi-Projection Acceleration
- New Antialiasing Features
- Misc other new features
- Questions and Answers
GEOMETRY SHADER CHALLENGES

- Significant overhead even for pass-through cases
- Significant overhead for viewport selection
- Significant amplification overhead for multiple viewports
MULTI-PROJECTION ACCELERATION

- Fast Geometry Shader pass-through
- Fast Viewport/RT multi-casting
- Maxwell accelerates:
  - Voxelization
  - Cube-map rendering
  - Cascaded shadow maps
  - Multi-resolution rendering

ViewportMask = 0b1101
MULTI-PROJECTION ACCELERATION

- Fast Geometry Shader pass-through
- Fast Viewport multi-casting

- Maxwell accelerates:
  - Voxelization
  - Cube-map rendering
  - Cascaded shadow maps
  - Multi-resolution rendering
MULTI-PROJECTION ACCELERATION

- Fast Geometry Shader pass-through
- Fast Viewport multi-casting

- Maxwell accelerates:
  - Voxelization
  - Cube-map rendering
  - Cascaded shadow maps
  - Multi-resolution rendering
MULTI-PROJECTION ACCELERATION

- Fast Geometry Shader pass-through
- Fast Viewport multi-casting
- Maxwell accelerates:
  - Voxelization
  - Cube-map rendering
  - Cascaded shadow maps
  - Multi-resolution rendering
MULTI-PROJECTION ACCELERATION

- Fast Geometry Shader pass-through
- Fast Viewport multi-casting

Maxwell accelerates:
  - Voxelization
  - Cube-map rendering
  - Cascaded shadow maps
  - Multi-resolution rendering
MULTI-PROJECTION API SUPPORT

- **OpenGL+Android:**
  - `NV_geometry_shader_passthrough` extension for GS pass-through
  - `NV_viewport_array2` extension for viewport multicast
  - The extension specs have good shader examples

- **DX11/DX12:**
  - No explicit API publicly available yet - stay tuned
OUTLINE

- Architectural goals of Maxwell
- DirectX12 hardware features
  - Conservative Rasterization
  - Raster Order Views
  - Tiled Resources
- Multi-Projection Acceleration
- New Antialiasing Features
- Misc other new features
- Questions and Answers
QUICK MULTISAMPLING RECAP
TARGET-INDEPENDENT RASTER

- Decouples visibility & raster rate from color sample rate
- Allows lower color buffer storage cost for custom AA techniques
- Introduces coverage reduction stage

\[
\frac{\text{CoverageAA}}{\text{DepthStencilAA}} \quad (\text{Eg. } 8x) \quad \geq \quad \text{ColorAA} \quad (\text{Eg. } 2x)
\]
POST-DEPTH COVERAGE

- Pre-Maxwell: Coverage Mask delivered is pre-depth-test coverage
  - No way to get at the post-depth-test coverage

- Maxwell can deliver post-depth-coverage to the pixel shader
SAMPLE COVERAGE OVERRIDE

- Pre-Maxwell: Shader can only reduce coverage sample set
- Maxwell can fully override raster-coverage mask
AGGREGATE G-BUFFER AA

- C. Crassin et al., "Aggregate G-Buffer Anti-Aliasing", ID3D 2015

- Uses post depth coverage to only process visible sub-samples

- Uses coverage override to route to right sub-sample cluster

- Other work using Maxwell AA features:
COVERAGE TO COLOR CONVERSION

Raster → Fragment Shader → Stencil/Depth-testing → Cvg. to color → Cvg. reduct. → Color Ops
PROGRAMMABLE SAMPLE LOCATIONS

- Sample locations fully programmable
- Interleaved sample positions
  - 16x sample locations can be tiled to a set of pixels
- Foundation for Multi Frame sampled AA
PROGRAMMABLE SAMPLE LOCATIONS

- Sample locations fully programmable
- Interleaved sample positions
  - 16x sample locations can be tiled to a set of pixels
- Foundation for Multi Frame sampled AA
AA FEATURES API SUPPORT

- OpenGL+ Android:
  - Target-independent multisampling control:
    - NV_framebuffer_mixed_samples
    - EXT_raster_multisample
  - Coverage to color conversion: NV_fragment_coverage_to_color
  - Post-depth coverage: EXT_post_depth_coverage
  - Multisample coverage override: NV_sample_mask_override_coverage
  - Programmable sample locations: NV_sample_locations

- DirectX FL 11.1
  - Target-independent multisampling

- DirectX 11 NvAPI:
  - NvAPI_D3D11_CreateRasterizerState
OUTLINE

- Architectural goals of Maxwell
- DirectX12 hardware features
  - Conservative Rasterization
  - Raster Order Views
  - Tiled Resources
- Multi-Projection Acceleration
- New Antialiasing Features
  - Misc other new features
- Questions and Answers
Screen Space Bonding Box rasterization
- Reduce # of vertices sent to GPU
- Speeds up particle systems, point sprite etc.

Attributes are extrapolated outside the primitive

Supported by these APIs:
- OpenGL: NV_fill_rectangle
- NvAPI: NvAPI_D3D11_CreateRasterizerState
MIN/MAX TEXTURE FILTERING

- Hardware support for min/max filtering

- Usecases:
  - Min-Max shadow maps
  - LOD maps for tiled textures
  - Other min-max reduction chains

- API support:
  - OpenGL: `EXT_texture_filter_minmax`
  - DirectX11.2

```plaintext
MAX returns “5”

MIN returns “0”
```
EXTENDED BLEND MODES

- ZERO_SRC
- DST
- SRC_OVER
- DST_OVER
- SRC_IN
- DST_IN
- SRC_OUT
- DST_OUT
- SRC_ATOP
- DST_ATOP
- XOR_PLUS
- PLUS_CLAMPED
- PLUS_CLAMPED_ALPHA
- MULTIPLY
- SCREEN
- OVERLAY
- DARKEN
- LIGHTEN
- COLOREDODGE
- COLORBURN
- HARDLIGHT
- SOFTLIGHT
- SOFTLIGHT_SVG
- DIFFERENCE
- MINUS
- MINUS_CLAMPED
- EXCLUSION
- CONTRAST
- INVERT_INVERT_RGB
- INVERT_KHR
- LINEARDODGE
- LINEARBURN
- VIVIDLIGHT
- LINEARLIGHT
- PINLIGHT
- HARDMIX
- RED
- GREEN
- BLUE
- HSL_HUE
- HSL_SATURATION
- HSL_COLOR
- HSL_LUMINOSITY

OpenGL: NV_blend_equation_advanced
FP16 ATOMIC OPERATIONS

- Vector 2x16-bit floating point atomic ADD, MIN, MAX
  - API supports 4x16-bit FP ops through 2 instructions

- Usecases:
  - Reduce the number of atomic ops during e.g. light accumulation
  - Save memory if you only need 16bit values

- API support:
  - OpenGL + Android: NV_shader_atomic_fp16_vector
  - NvAPI HLSL backdoor (described later):
    
    NvInterlocked[Add,Min,Max]Fp16x2(UAV, address, float2 value)
    NvInterlocked[Add,Min,Max]Fp16x4(UAV, address, float4 value)
Provides access to various new features from DX11 HLSL

Host part:

NvAPI_Initialize();
NvAPI_D3D11_SetNvShaderExtnSlot(7); // enable the backdoor on UAV 7 for example
pD3DDevice->Create{Pixel,Compute...}Shader(...);
NvAPI_D3D11_SetNvShaderExtnSlot(~0u); // disable the backdoor
// Call NvAPI_D3D11_IsNvShaderExtnOpCodeSupported(...) to test feature support

Shader part:

#define NV_SHADER_EXTN_SLOT u7 // must match the slot used above
#include “NvHlslExtns.h”

Then call the functions defined in that header.
OTHER HLSL FUNCTIONS

- **FP32 atomic ADD (Kepler+):**
  - `NvInterlockedAddFp32(UAV, address, float value)`

- **Warp shuffle (Kepler+):**
  - `NvShfl, NvShflUp, NvShflDown, NvShflXor(value, srcLane, width)`

- **Other warp-synchronous functions (Fermi+):**
  - `NvAny, NvAll, NvBallot(predicate)`
  - `NvGetLaneId()`

- Warp-synchronous functions work in pixel shaders too
OUTLINE

- Architectural goals of Maxwell
- DirectX12 hardware features
  - Conservative Rasterization
  - Raster Order Views
  - Tiled Resources
- Multi-Projection Acceleration
- New Antialiasing Features
- Misc other new features
- Questions and Answers
THANK YOU!

JOIN THE CONVERSATION
#GTC15