GPU (Graphics Processing Unit) with a Focus on Nvidia GeForce 6 Series

By:
- Binesh Tuladhar
- Clay Smith
Overview

- History of GPU’s
- GPU Definition
- Classical Graphics Pipeline
- Geforce 6 Series Architecture
- Vertex and Fragment Processors
- Traditional Architecture vs Super Scalar Architecture
- GPU features – Functional and Shader Model 3.0
- Fragment Processor Performance
- Future of GPU’s
History

- Early GPU’s
  - designed for primitive graphics operations

- 1990’s
  - 2D graphics accelerators evolved

- Mid 1990’s – late 1990’s
  - CPU assisted 3D graphics for PC’s and gaming consoles
  - 3D hardware supporting T&L

- Modern GPU’s
  - Programmable Graphics Pipeline with pixel and vertex shaders
GPU Definition

- Graphics rendering device
Definition contd...

- Consists of processor which is used for floating point calculations and rendering of 2D/3D images

- Processors either attached to graphics card or integrated in system board

- Decreases work load of CPU

- Gives improved system performance and gives realistic effect to images, videos and games
Classical Graphics Pipeline

- **Vertex**
  - T & L evolved to vertex

- **Triangle**
  - Triangle, point, line setup

- **Pixel**
  - Flat shading, texturing, eventually pixel shading

- **ROP**
  - Blending, Z buffering, antialiasing

- **Memory**
  - Wider and faster over years
Definition Contd...

- GPU interfaces with the CPU using fast buses like AGP (uptp 2GB/sec for 8X AGP)

- PCI Express (upto 8GB/sec)

- Bus speeds are important because textures and vertex data needs to come from CPU to GPU
GeForce 6 Series Architecture
GeForce 6 Series Vertex Processor

- Vertex data received from host CPU
- Allow programs that performs transformation, skinning and other per-vertex operations.
- All operations are 32 bit floating point (fp32) precision per component
- Vertex cache present- stores vertex data
- These vertices are grouped inty primitives- points, lines, triangles
Cull/Clip, Rasterization and Z-Cull

- **Cull/Clip**
  - Removes invisible primitives and performs plane equation on data for rasterization

- **Rasterization**
  - Computes pixel coverage and uses Z-cull to discard pixels blocked by objects

- **Z-Cull**
  - Eliminates pixel based on the depth
GeForce 6 Series Fragment Processor

- Fragment – a candidate pixel
- Texture and fragments units operates on quads (squares of 4 pixels)
- Works on groups of 100 of pixels at a time in SIMD fashion
- Uses texture caches or units to fetch data from memory
Fragment Processor Contd...

- Shader units can perform 8 math ops (w/o texture fetch) or 4 math ops (with texture fetch) in each clock cycle

- Fog calculation done in the end

\[ \text{out} = \text{FogColor} \times \text{fogFraction} + \text{SrcColor} \times (1-\text{fogFraction}) \]

- Pixels almost ready for framebuffer
Z-Compare and Blending

Fragments are passed through Z-compare and Blending units

Operations:

- Depth testing
- Stencil tests
- Alpha operations
- Final color write to target surface
Memory

- Divided into independent partitions each with its own DRAM’s
- Independent partitions increase latency and efficiency regardless of size of data transferred
- Gives GPU a wide 256 bits of flexible memory subsystem
- Allows streaming for small memory accesses (32 bytes) at 35 GB/sec
- In low end system, system memory shared as graphics memory
Traditional Architecture

- Traditional non scalar shader architecture has only one shader unit.
- Can process only upto 4 operations per cycle.
Super-scalar Architecture

- superscalar architecture has a second shader unit
- doubles pixel operations per cycle
- Can process up to 8 operations per cycle
GPU Features

Functional Features

- Geometric Instancing
  - Vertex stream frequency
    - hardware support for looping over a subset of vertices

Example: rendering the same object multiple times at different locations (grass, soldiers, people in stadium)
GPU Features Contd...

- Early culling and clipping
  - remove nonvisible primitives at high rate

- Rasterization
  - rendering supports
    - point sprites
    - Aliased and anti-aliased lines
    - Aliased and antialiased triangles

- Z-Cull
  - Allows high-speed removal of hidden surfaces

- Occlusion Query
  - Keeps a record of the number of fragments passing or failing the depth test and reports it to the CPU
GPU Features Contd...

- **Texturing**
  - Textures can now directly be fetched into the vertex program.

- **Shadow Buffer Support**
  - Fetches shadow buffer as a projective texture and performs $z$ compares of the shadow buffer data to distance from light.
Shader Model 3.0 Features

- Increased instruction count (up to 65535 instructions.)
- Fragment processor; multiple render targets.
- Dynamic flow control branching
- Vertex texturing
- More temporary registers.
Shader Model 3.0 Features Contd..

- **Co-issue**
  - Two separate operations can concurrently execute on different parts of a four-wide register

- **Dual Issue**
  - Independent instructions can be executed on independent units in computational pipeline
Fragment Processor Performance

- fp32 and fp16 precision support for intermediate calculations

- Ability to perform four wide, coissue multiply-add (MAD) or four-term dot product (DP4), plus a four-wide, coissue-able and dual-issuable multiply instruction per clock in series.

- Dedicated fp16 normalization hardware
Future of GPU’s

- High end graphics card to support graphical applications games requiring high resolutions and faster refresh rates
- Increase speeds of hardware such as memory and processor
- More compact, low power, low cost GPU’s
- Empirical testing of infield operation
Bibliography


