Graphics Hardware

CS7GV3 – Real-time Rendering
Digital Logic and Arithmetic

• Ripple Carry Adder
Function Unit

Fundamental Micro-operations

• Register transfer
• Arithmetic operations
  • Addition
  • Subtraction
• Logic operations
  • AND, OR, XOR ...
• Shift operations
  • Left, Right ...
Floating-point Numbers

\[ X = \pm m \cdot b^e \]

\[
\begin{array}{cccc}
  & & k & 0 \\
  s & \text{exponent} & \text{mantissa} \\
\end{array}
\]

\( b = 2 \)
\( e = \text{exponent} - \text{bias} \)
\( \text{bias} = 2^{j-k-1}-1 \)
\( \text{if } (m \neq 0) \text{ then } m = 1.\text{mantissa} \)
\( \text{if } (s = 0) \text{ then } X > 0 \)
\( \text{if } (s = 1) \text{ then } X < 0 \)

IEEE Specification


![Diagram](image)

<table>
<thead>
<tr>
<th>Precision</th>
<th>I</th>
<th>j</th>
<th>k</th>
<th>e(bits)</th>
<th>m(bits)</th>
<th>bias</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single</td>
<td>31</td>
<td>30</td>
<td>22</td>
<td>8</td>
<td>24</td>
<td>$2^{127}=2^7-1$</td>
</tr>
<tr>
<td>Double</td>
<td>63</td>
<td>62</td>
<td>51</td>
<td>11</td>
<td>53</td>
<td>$2^{1023}=2^{10}-1$</td>
</tr>
</tbody>
</table>

**Single**  \((-1)^s \times (1.m) \times 2^{e-127}\)

**Double**  \((-1)^s \times (1.m) \times 2^{e-1023}\)
Exponent Compare & Align the Mantissa

\[ X = 1.1100 \times 2^4 = 28 \]
\[ +Y = 1.1000 \times 2^2 = 6 \]

Align
\[
\begin{array}{c}
1.1100 \times 2^4 \\
+0.0110 \times 2^4 \\
10.0010 \times 2^4 = 34
\end{array}
\]
Floating-point Adder

Pipelined Floating-point Adder

Datapath
Algorithmic State Machine

Control for the Datapath
Xbox 360 (2005)

- **CPU**
  - 3 cores

- **GPU**
  - 48 parallel, unified shaders
Amdahl’s Law

• Amdahl’s Law equation (1) allows us to compute the speedup that can be achieved.

\[
\text{Speedup} = \frac{1}{\frac{\text{Fraction}_{\text{enhanced}}}{\text{Speedup}_{\text{enhanced}}} + (1 - \text{Fraction}_{\text{enhanced}})}
\]  

(1)

• Amdahl’s Law equation (2) tells us that in order to achieve a speedup of 80 with 100 processors only 0.25% of the code may executed sequentially.

\[
80 = \frac{1}{\frac{\text{Fraction}_{\text{parallel}}}{100} + (1 - \text{Fraction}_{\text{parallel}})}
\]  

(2)


- Streaming-processors (SPs)
  - 128 (5120 in 2018)
- Streaming-multiprocessors (SMs)
  - 16
- Texture Processor Clusters (TPCs)
  - 8
- No logical order of graphics pipeline stages

Scalable Memory System

- **Raster operation-processors (ROPs)**
  - 6 (fixed-function)
  - Perform color and depth frame buffer directly on memory

- **Interconnection Network**
  - Moves fragment colour and depth from Streaming processor array (SPA) to Raster operation-processors (ROPs)
  - Also provides access to the textures in the external DRAM

---

Input to the Streaming processor array (SPA)

- **Input assembler**
  - Collects vertex work as directed by the input command stream

- **Vertex work distribution**
  - Distributes Vertex work packets to TPCs in the SPA

- **The Texture Processor Clusters (TPCs) executes:**
  - Vertex shader programs
  - Geometry shader programs
  - Output is written to on-chip buffers

- **Viewport/clip/setup/raster/Zcull**
  - Takes the output from the buffer
  - Rasterizes it into fragments

- **Pixel work distribution**
  - Send fragments to the appropriate TCDs
Texture Processor Cluster (TPC)

- Geometry Controller
  - Manages input and output vertex attributes
- SM Controller (SMC)
- Streaming multiprocessors (SMs)
  - Two
- Texture Unit
Streaming Multiprocessor (SM)

- Unified graphics and computing multiprocessor
  - Vertex shader
  - Geometry shader
  - Fragment shader
  - Parallel computing shader
- Special function Units (SFUs)
- Multithreaded instruction fetch and issue unit (MT Issue)

Single-instruction Multiple-thread

- Warps
  - 32 parallel threads
  - Pool of 24 warps
  - Total of 768

SIMT Stack

foo[] = {4,8,12,16};

A: v = foo[tid.x];
B: if (v < 10)
C: v = 0;
    else
D: v = 10;
E: w = bar[tid.x]+v;

Handles **Branch Divergence**

One stack per warp

<table>
<thead>
<tr>
<th>PC</th>
<th>RPC</th>
<th>Active Mask</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td>-</td>
<td>1111</td>
</tr>
<tr>
<td>D</td>
<td>E</td>
<td>0011</td>
</tr>
<tr>
<td>C</td>
<td>E</td>
<td>1100</td>
</tr>
</tbody>
</table>
GeForce 8800 Ultra Die Layout

- Text

NVIDIA TITAN (2018)

• Transistor Count
  • 21.1 Billion
• Total Video Memory
  • 12288 MB HBM2
• CUDA Cores
  • 5120 (single precision)
• 640 Tensor Cores
• Streaming Multiprocessors (SMs)
  • 80 (64 SP per SM)
• 110 TeraFLOPS