Reading Assignment #3 (until Feb. 24)

Read (required):

• Programming Massively Parallel Processors book, Chapter 1 (Introduction)
• Programming Massively Parallel Processors book, Appendix B (GPU Compute Capabilities)
• GLSL book, Chapter 6 (Simple Shading Example)
• GLSL book, Chapter 8.1-8.3 (Shader Development)

Read (optional):

• GLSL book, Chapter 7 (OpenGL Shading Language API)
  You will need some of this information for programming assignment #2!
My chip!

16 cores

8 mul-add ALUs per core
(128 total)

16 simultaneous
instruction streams

64 concurrent (but interleaved)
instruction streams

512 concurrent fragments

= 256 GFLOPs  (@ 1GHz)
My “enthusiast” chip!

32 cores, 16 ALUs per core (512 total) = 1 TFLOP (@ 1 GHz)
Summary: three key ideas for high-throughput execution

1. Use many “slimmed down cores,” run them in parallel

2. Pack cores full of ALUs (by sharing instruction stream overhead across groups of fragments)
   - Option 1: Explicit SIMD vector instructions
   - Option 2: Implicit sharing managed by hardware

3. Avoid latency stalls by interleaving execution of many groups of fragments
   - When one group stalls, work on another group
NVIDIA GeForce GTX 480 “core”

- Groups of 32 fragments share an instruction stream
- Up to 48 groups are simultaneously interleaved
- Up to 1536 individual contexts can be stored

Source: Fermi Compute Architecture Whitepaper
CUDA Programming Guide 3.1, Appendix G
NVIDIA GeForce GTX 480 “core”

- **Fetch/Decode**
- **Execution contexts** (128 KB)
- **“Shared” scratchpad memory** (16+48 KB)

SIMD function unit, control shared across 16 units (1 MUL-ADD per clock)

- The core contains 32 functional units
- Two groups are selected each clock (decode, fetch, and execute two instruction streams in parallel)

Source: Fermi Compute Architecture Whitepaper
CUDA Programming Guide 3.1, Appendix G

Kayvon Fatahalian, Graphics and Imaging Architectures (CMU 15-669, Fall 2011)
**NVIDIA GeForce GTX 480 “SM”**

- **Fetch/Decode**
- **Execution contexts (128 KB)**
- **“Shared” scratchpad memory (16+48 KB)**

= CUDA core  
(1 MUL-ADD per clock)

- The SM contains 32 CUDA cores
- Two warps are selected each clock (decode, fetch, and execute two warps in parallel)
- Up to 48 warps are interleaved, totaling 1536 CUDA threads

Source: Fermi Compute Architecture Whitepaper  
CUDA Programming Guide 3.1, Appendix G

---

Kayvon Fatahalian, Graphics and Imaging Architectures (CMU 15-869, Fall 2011)
NVIDIA GeForce GTX 480

There are 15 of these things on the GTX 480:
That’s 23,000 fragments!
(or 23,000 CUDA threads!)
Bonus slides: NVIDIA GTX 680 (2012)

NVIDIA Kepler GK104 architecture SMX unit (one "core")

Warp execution contexts (256 KB)

"Shared" memory or L1 data cache (64 KB)

= SIMD function unit, control shared across 32 units (1 MUL-ADD per clock)

= "special" SIMD function unit, control shared across 32 units (operations like sin/cos)

= SIMD load/store unit (handles warp loads/stores, gathers/scatters)

CMU 15-418, Spring 2013
Bonus slides: NVIDIA GTX 680 (2012)

NVIDIA Kepler GK104 architecture SMX unit (one “core”)

- SMX core resource limits:
  - Maximum warp execution contexts: 64 (2,048 total CUDA threads)
  - Maximum thread blocks: 16

- SMX core operation each clock:
  - Select up to four runnable warps from up to 64 resident on core (thread-level parallelism)
  - Select up to two runnable instructions per warp (instruction-level parallelism)
  - Execute instructions on available groups of SIMD ALUs, special-function ALUs, or LD/ST units
**Bonus slides: NVIDIA GTX 680 (2012)**

*NVIDIA Kepler GK104 architecture*

- 1 GHz clock
- Eight SMX cores per chip
- $8 \times 192 = 1,536$ SIMD mul-add ALUs
  - $= 3$ TFLOPs
- Up to 512 interleaved warps per chip
  - (16,384 CUDA threads/chip)
- TDP: 195 watts

---

**Memory**

256 bit interface

DDR5 DRAM

---

192 GB/sec
AMD Radeon HD 5870 (Cypress)

- **AMD-speak:**
  - 1600 stream processors

- **Generic speak:**
  - 20 cores
  - 16 “beefy” SIMD functional units per core
  - 5 multiply-adds per functional unit (VLIW processing)
**AMD Radeon HD 5870 “core”**

**Groups of 64 [fragments/vertices/etc.] share instruction stream**

**Four clocks to execute an instruction for all fragments in a group**

---

Source: ATI Radeon HD5000 Series: An Inside View (HPG 2010)
AMD Radeon HD 5870 “SIMD-engine”

Groups of 64 [fragments/vertices/OpenCL work items] are in a “wavefront”.

Four clocks to execute an instruction for an entire wavefront

Source: ATI Radeon HD5000 Series: An Inside View (HPG 2010)
AMD Radeon HD 5870

There are 20 of these “cores” on the 5870: that’s about 31,000 fragments!
• Streaming Processor (SP)
• Streaming Multiprocessor (SM)
• Texture/Processing Cluster (TPC)

Courtesy AnandTech
NVIDIA G80/GT200 Architecture

- G80/G92: \(8 \text{ TPCs} \times (2 \times 8 \text{ SPs}) = 128 \text{ SPs}\)
- GT200: \(10 \text{ TPCs} \times (3 \times 8 \text{ SPs}) = 240 \text{ SPs}\)

- Arithmetic intensity has increased (ALUs vs. texture units)
Thank you.

Thanks for slides

• Kayvon Fatahalian