CSCI-GA.3033-004

Graphics Processing Units (GPUs): Architecture and Programming

Lecture 1: Gentle Introduction to GPUs

Mohamed Zahran (aka Z)
mzahran@cs.nyu.edu
http://www.mzahran.com
Who Am I?

- Mohamed Zahran (aka Z)
- Computer architecture/OS/Compilers Interaction
- http://www.mzahran.com
- Office hours: Wed/Th 2:00-3:00 pm
  - Or by appointment
- Room: WWH 320
*Formal Goals of This Course*

- Why GPUs
- GPU Architecture
- GPU-CPU Interaction
- GPU programming model
- Solving real-life problems using GPUs
  - With the best performance we can get!
Informal Goals of This Course

• To get more than an A
• To use what you have learned in MANY different contexts
• To have a feeling about how hardware and software evolve and interact
• To enjoy the course!
The Course Web Page

• Lecture slides
• Reading assignments
• Info about mailing list, labs, homework assignments, project, and exam.
• Useful links (manuals, tools, book errata, ... )
The Textbook

Programming Massively Parallel Processors: A Hands-on Approach

By

David B. Kirk & Wen-mei W. Hwu

2nd Edition
Grading

• Homework assignments : 15%
• Project : 25%
• Programming assignments : 30%
• Final (open book/notes) : 30%
Computer History

Eckert and Mauchly

- 1st working electronic computer (1946)
- 18,000 Vacuum tubes
- 1,800 instructions/sec
- 3,000 ft³
Computer History

- Maurice Wilkes

EDSAC 1 (1949)

1st stored program computer
650 instructions/sec
1,400 ft³

http://www.cl.cam.ac.uk/UoCCL/misc/EDSAC99/
Intel 4004 Die Photo

- Introduced in 1970
  - First microprocessor
- 2,250 transistors
- 12 mm²
- 108 KHz
Intel 8086 Die Scan

- 29,000 transistors
- 33 mm$^2$
- 5 MHz
- Introduced in 1979
  - Basic architecture of the IA32 PC
Intel 80486 Die Scan

- 1,200,000 transistors
- 81 mm²
- 25 MHz
- Introduced in 1989
  - 1st pipelined implementation of IA32
Pentium Die Photo

- 3,100,000 transistors
- 296 mm²
- 60 MHz
- Introduced in 1993
  - 1st superscalar implementation of IA32
Pentium III

- 9,500,000 transistors
- 125 mm²
- 450 MHz
- Introduced in 1999

Pentium 4

- 55,000,000 transistors
- 146 mm²
- 3 GHz
- Introduced in 2000

http://www.chip-architect.com
The Famous Moore’s Law
People ask for more improvements → Hardware Improvement → Better Software → People get used to the software

Positive Cycle of Computer Industry
The Status-Quo

• We moved from single core to multicore
  – for technological reasons
• Free lunch is over for software folks
  – The software will not become faster with every new generation of processors
• Not enough experience in parallel programming
  – Parallel programs of old days were restricted to some elite applications -> very few programmers
  – Now we need parallel programs for many different applications
Two Main Goals

• Maintain execution speed of old sequential programs

• Increase throughput of parallel programs
Two Main Goals

• Maintain execution speed of old sequential programs

• Increase throughput of parallel programs
Performance

Theoretical GFLOPS/s

Source: NVIDIA CUDA C Programming Guide
CPU is optimized for sequential code performance
Almost 10x the bandwidth of multicore (relaxed memory model)
Source: NVIDIA CUDA C Programming Guide
How to Choose A Processor for Your Application?

- Performance
- Very large installation base
- Practical form-factor and easy accessibility
- Support for IEEE floating point standard
Integrated GPU vs Discrete GPU

(a) and (b) represent discrete GPU solutions, with a CPU-integrated memory controller in (b). Diagram (c) corresponds to integrated CPU-GPU solutions, as the AMD's Accelerated Processing Unit (APU) chips.

Tradeoff: Low energy vs higher performance

Integrated CPU+GPU processors

- **More than 90%** of processors shipping today include a GPU on die
- **Low energy use** is a key design goal

Intel 4th Generation Core Processor: “Haswell”

AMD Kaveri APU

![Haswell Processor Die Map](http://www.geeks3d.com/20140114/amd-kaveri-a10-7850k-a10-7700k-and-a8-7600-apus-announced/)

---

**4-core GT2 Desktop:** 35 W package  
**2-core GT2 Ultrabook:** 11.5 W package

**Desktop:** 45-95 W package  
**Mobile, embedded:** 15 W package

source: Performance and Programmability Trade-offs in the OpenCL 2.0 SVM and Memory Model by Brian T. Lewis, Intel Labs
Is Any Application Suitable for GPU?

• Heck no!

• You will get the best performance from GPU if your application is:
  – Computation intensive
  – May independent computations
  – Many similar computations

- 16 highly threaded SM’s,
- >128 FPU’s, 367 GFLOPS,
- 768 MB DRAM,
- 86.4 GB/S Mem BW,
- 4GB/S BW to CPU
A Glimpse at A Modern GPU

Streaming Multiprocessor (SM)
A Glimpse at A Modern GPU

Streaming Processor (SP)

SPs within SM share control logic and instruction cache
A Glimpse at A Modern GPU

- Much higher bandwidth than typical system memory
- A bit slower than typical system memory
- Communication between GPU memory and system memory is slow
Amdahl's Law

Execution Time After Improvement =
Execution Time Unaffected + ( Execution Time Affected / Amount of Improvement )

• Example:

"Suppose a program runs in 100 seconds on a machine, with multiply responsible for 80 seconds of this time. How much do we have to improve the speed of multiplication if we want the program to run 4 times faster?"

How about making it 5 times faster?

Improvement in your application speed depends on the portion that is parallelized
Things to Keep in Mind

• Try to increase the portion of your program that can be parallelized
• Figure out how to get around limited bandwidth of system memory
• When an application is suitable for parallel execution, a good implementation on GPU can achieve more than 100x speedup over sequential implementation.
• You can reach 10x fairly easy, beyond that ... stay with us!
Enough for Today

• Some applications are better run on CPU while others on GPU
• If you don’t care about performance, parallel programming is easy!
• Main limitations
  – The parallelizable portion of the code
  – The communication overhead between CPU and GPU
  – Memory bandwidth saturation

Welcome ... And Have Fun!