The background of the slide is a dark orange color with a white circuit board pattern. The pattern consists of numerous parallel lines and various shapes, including circles and rectangles, representing traces and components on a PCB. The lines are arranged in a somewhat regular, grid-like fashion on the left side, becoming more irregular and branching out towards the right side.

**Heterogeneous
Parallel
Programming**

Lecture 1.2

**Introduction to Heterogeneous
Parallel Computing**

Wen-mei Hwu - University of Illinois at Urbana-Champaign

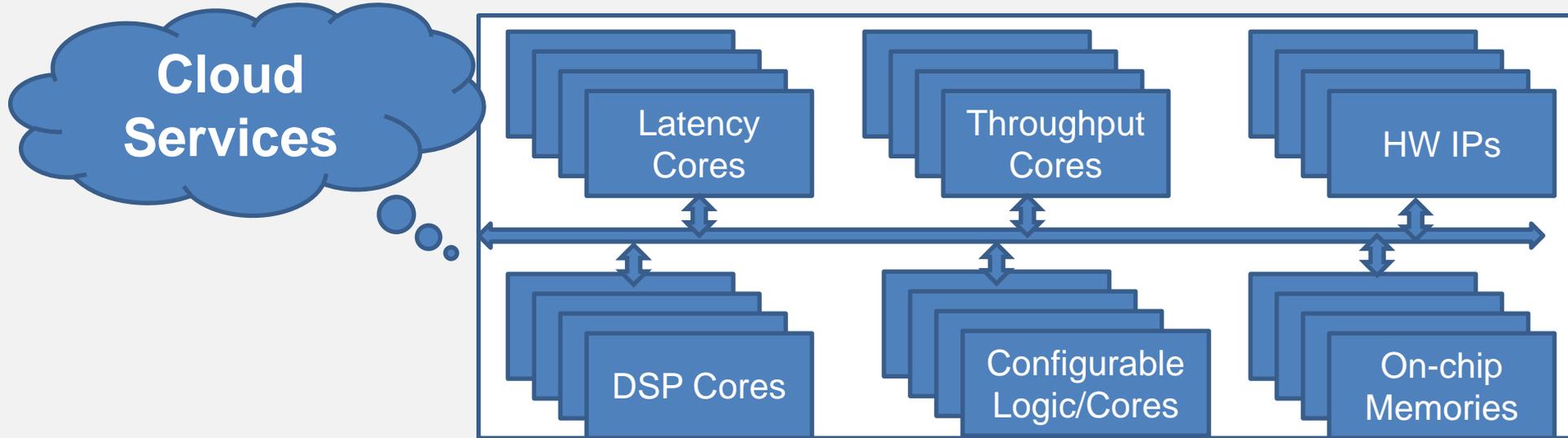


Objectives

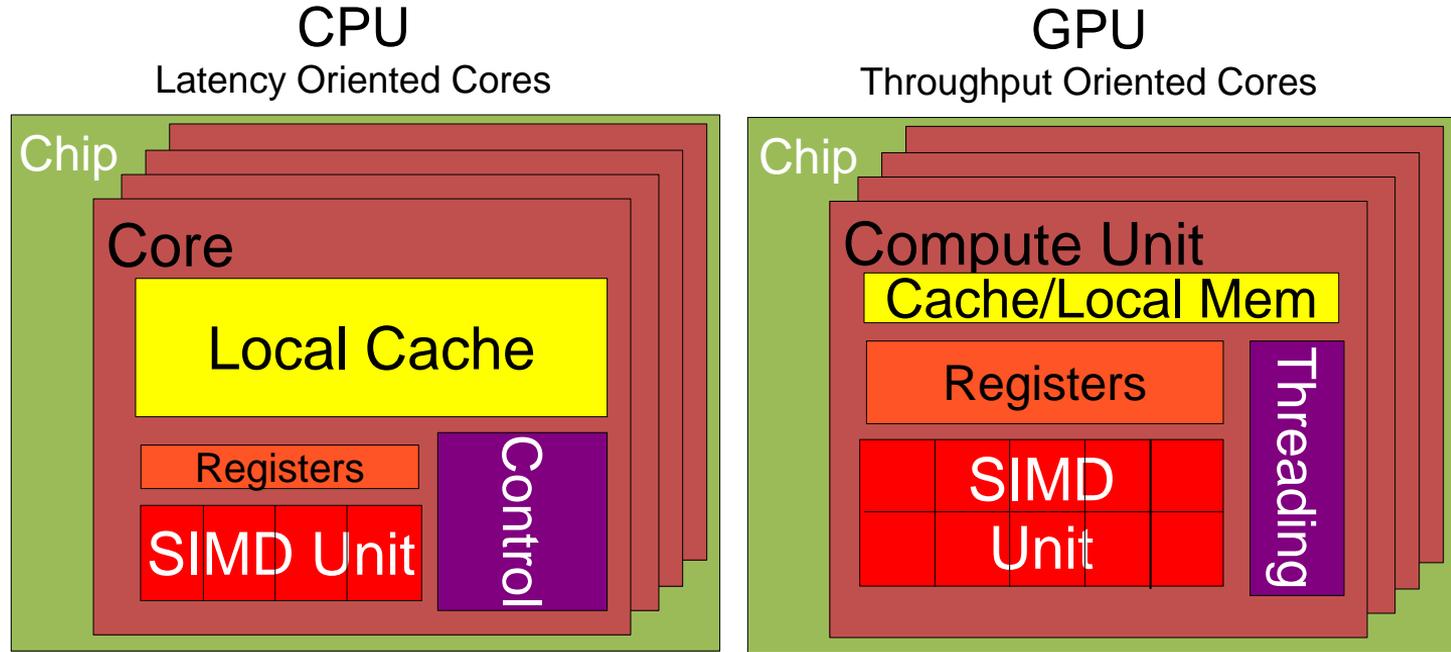
- To learn the major differences between latency devices (CPU cores) and throughput devices (GPU cores)
- To understand why winning applications increasingly use both types of devices

Heterogeneous Parallel Computing

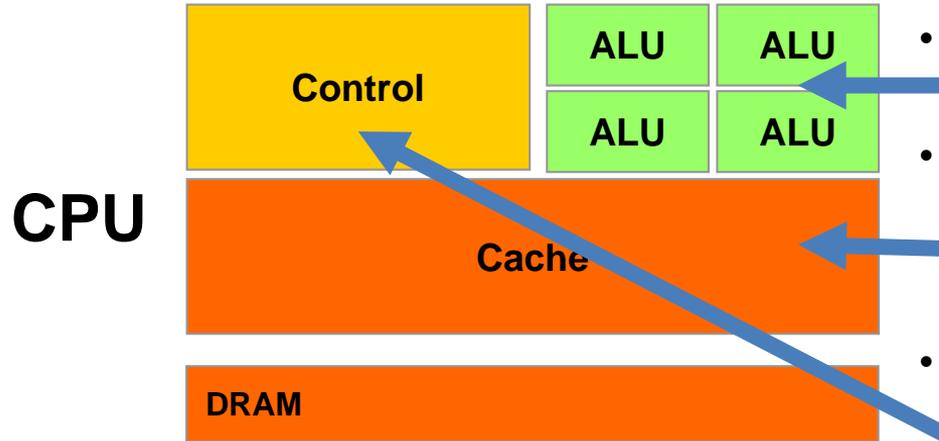
- Use the best match for the job (heterogeneity in mobile SOC)



CPU and GPU are designed very differently



CPUs: Latency Oriented Design



- Powerful ALU
 - Reduced operation latency
- Large caches
 - Convert long latency memory accesses to short latency cache accesses
- Sophisticated control
 - Branch prediction for reduced branch latency
 - Data forwarding for reduced data latency

GPUs: Throughput Oriented Design

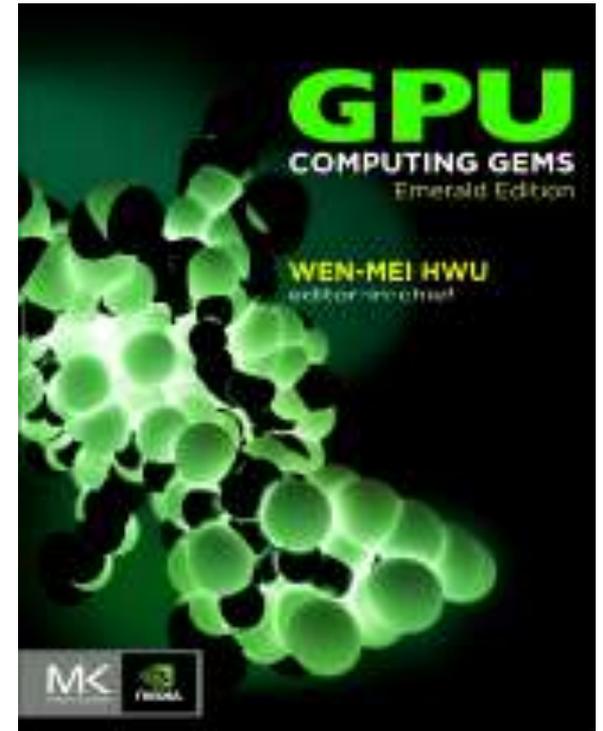
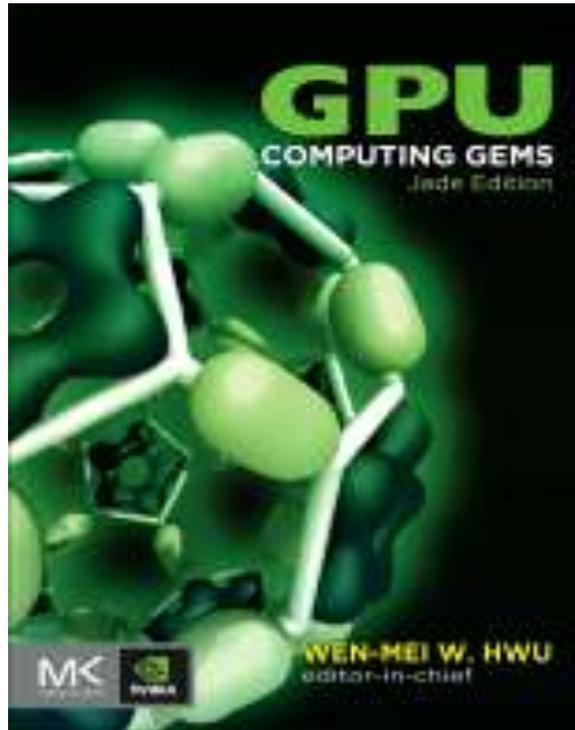


- Small caches
 - To boost memory throughput
- Simple control
 - No branch prediction
 - No data forwarding
- Energy efficient ALUs
 - Many, long latency but heavily pipelined for high throughput
- Require massive number of threads to tolerate latencies

Winning Applications Use Both CPU and GPU

- CPUs for sequential parts where latency matters
 - CPUs can be 10+X faster than GPUs for sequential code
- GPUs for parallel parts where throughput wins
 - GPUs can be 10+X faster than CPUs for parallel code

GPU computing is catching on.



90 articles in two volumes

Heterogeneous parallel computing is catching on.

Financial
Analysis

Scientific
Simulation

Engineering
Simulation

Data
Intensive
Analytics

Medical
Imaging

Digital
Audio
Processing

Digital
Video
Processing

Computer
Vision

Biomedical
Informatics

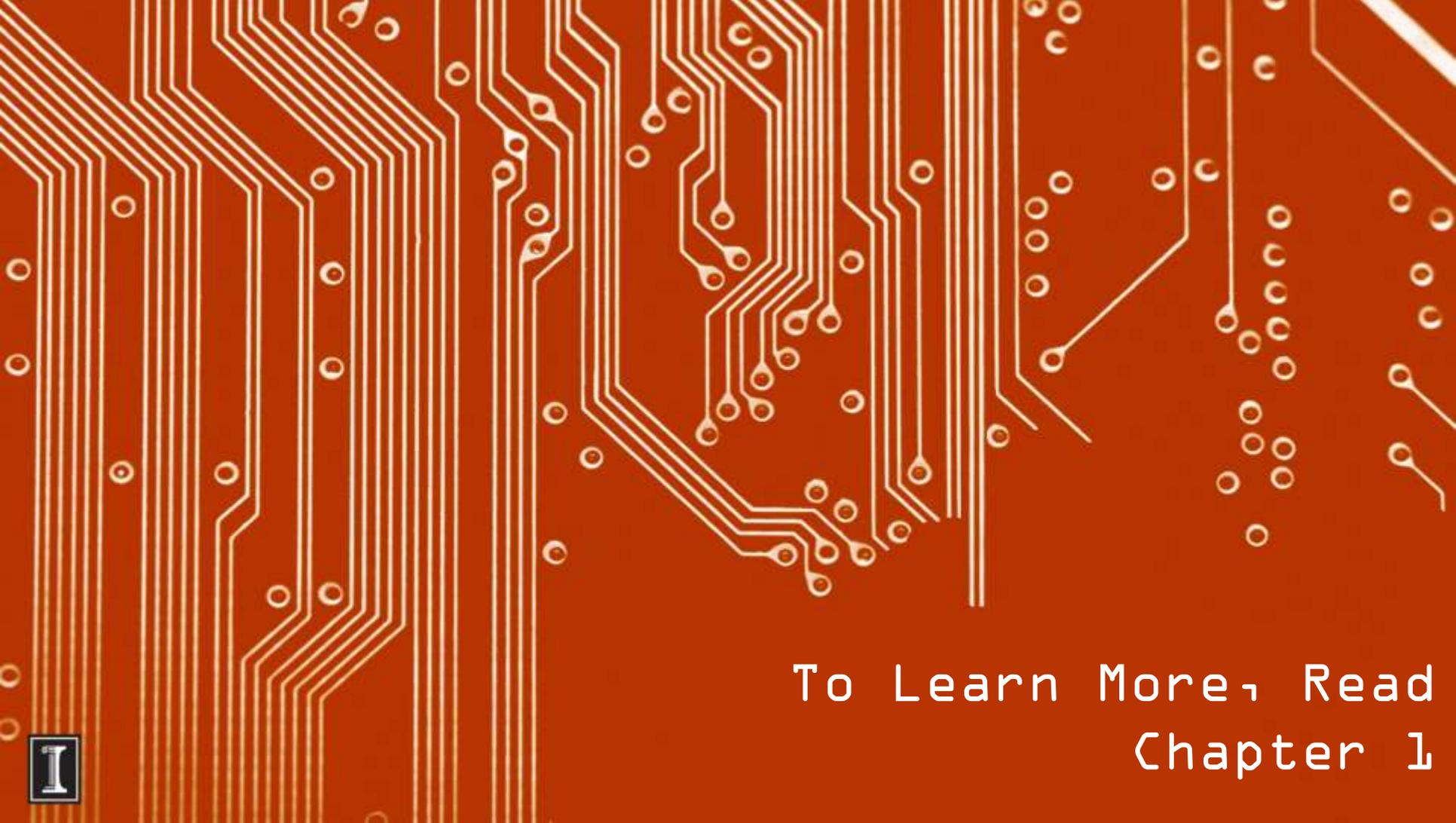
Electronic
Design
Automation

Statistical
Modeling

Numerical
Methods

Ray Tracing
Rendering

Interactive
Physics



To Learn More, Read
Chapter 1

