CS 181AI
Lecture 11

CPU vs. GPU

Arthi Padmanabhan

Feb 22, 2023

Logistics

- Assignment 3 due Friday
  - No formal office hours today but my door will be open this afternoon from 2 - 4

Last Time

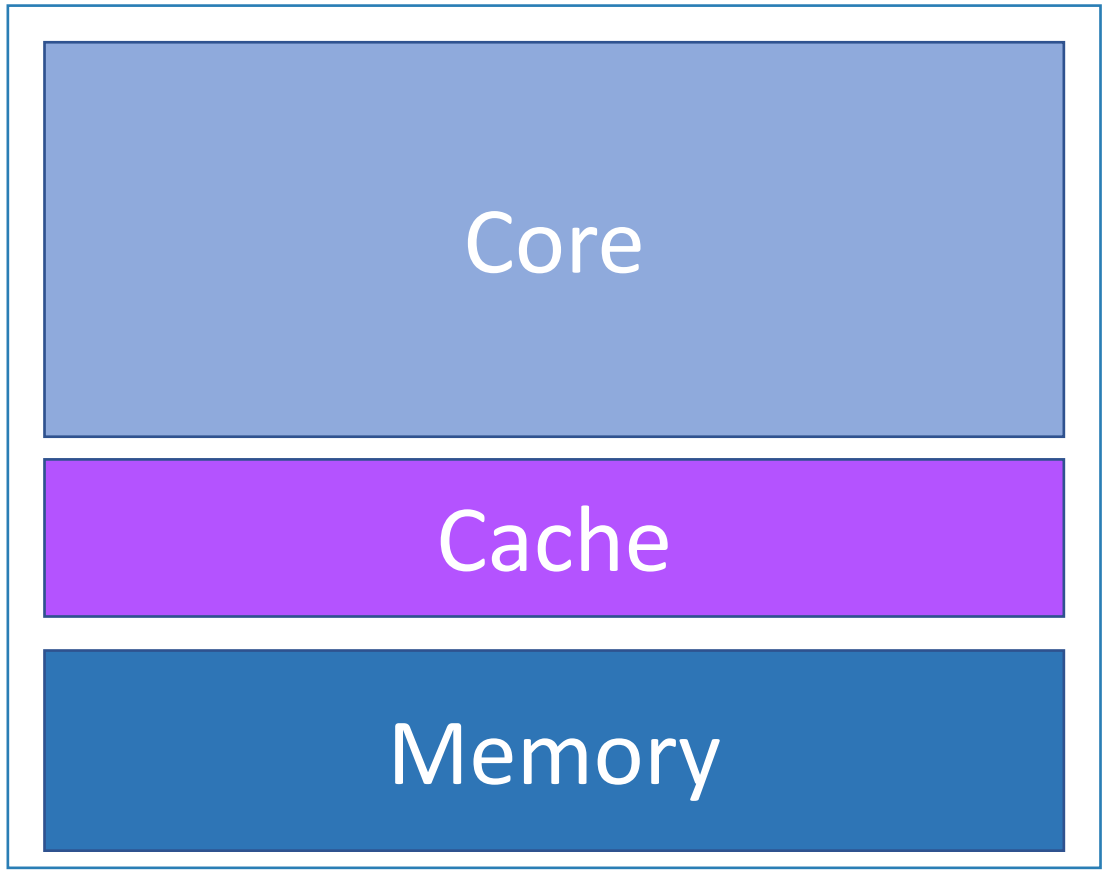
- Finished Synchronization
  - Monitors & conditional variables

Today

- GPUs!

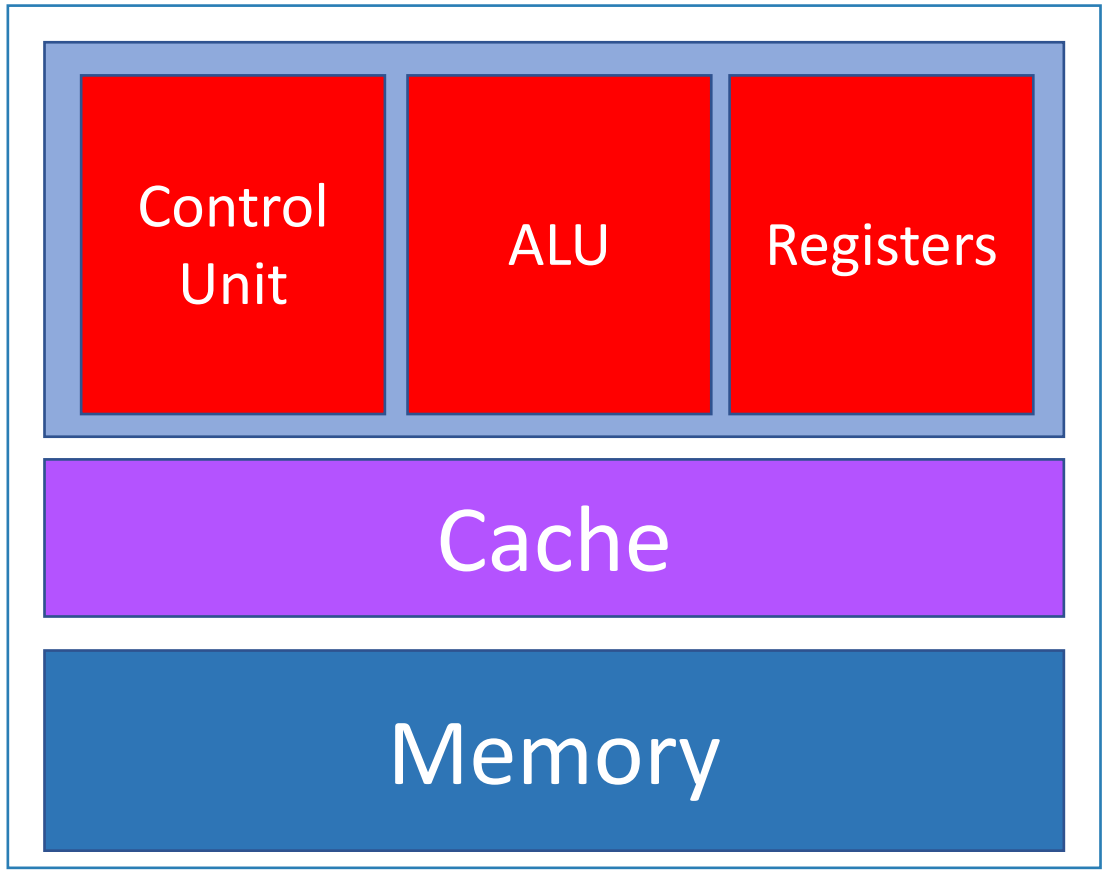
CPU architecture

- CPU is designed to perform a large variety of tasks



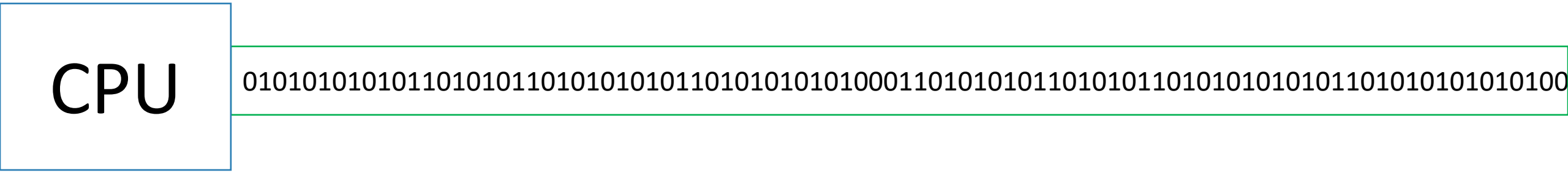
CPU architecture

- CPU is designed to perform a large variety of tasks



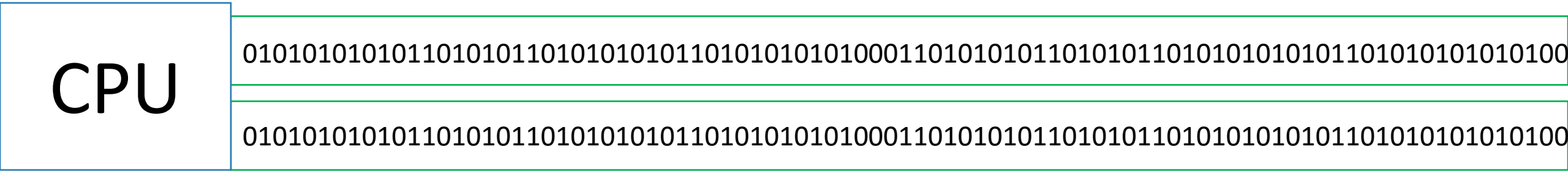
CPU Threads

- A CPU core can only process one instruction at any given time



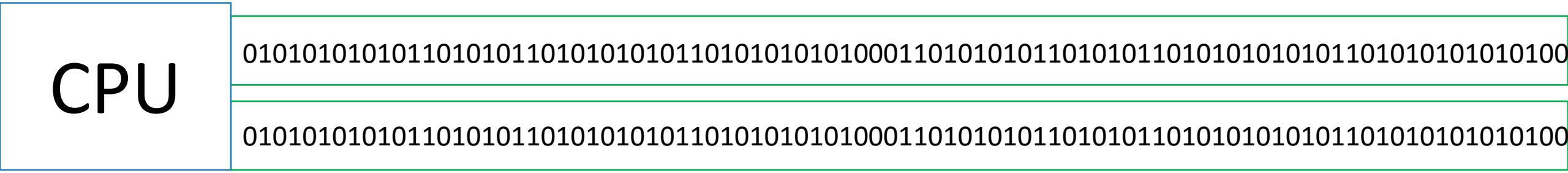
CPU Threads

- CPU cores only process one instruction at any given time
- Can have multiple threads though



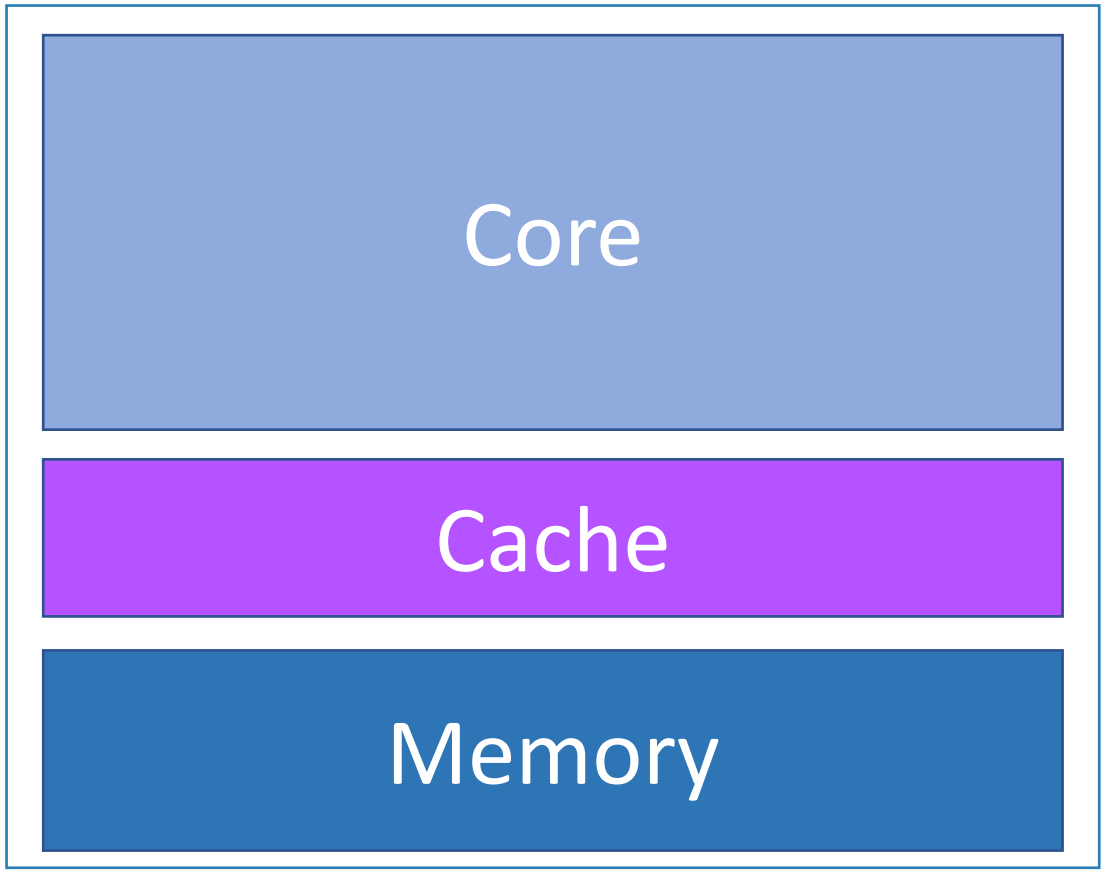
CPU Threads

- CPU cores only process one instruction at any given time
- Can have multiple threads though
- Threads within a CPU core can run concurrently (not in parallel)
- So nothing can happen in parallel on a CPU?



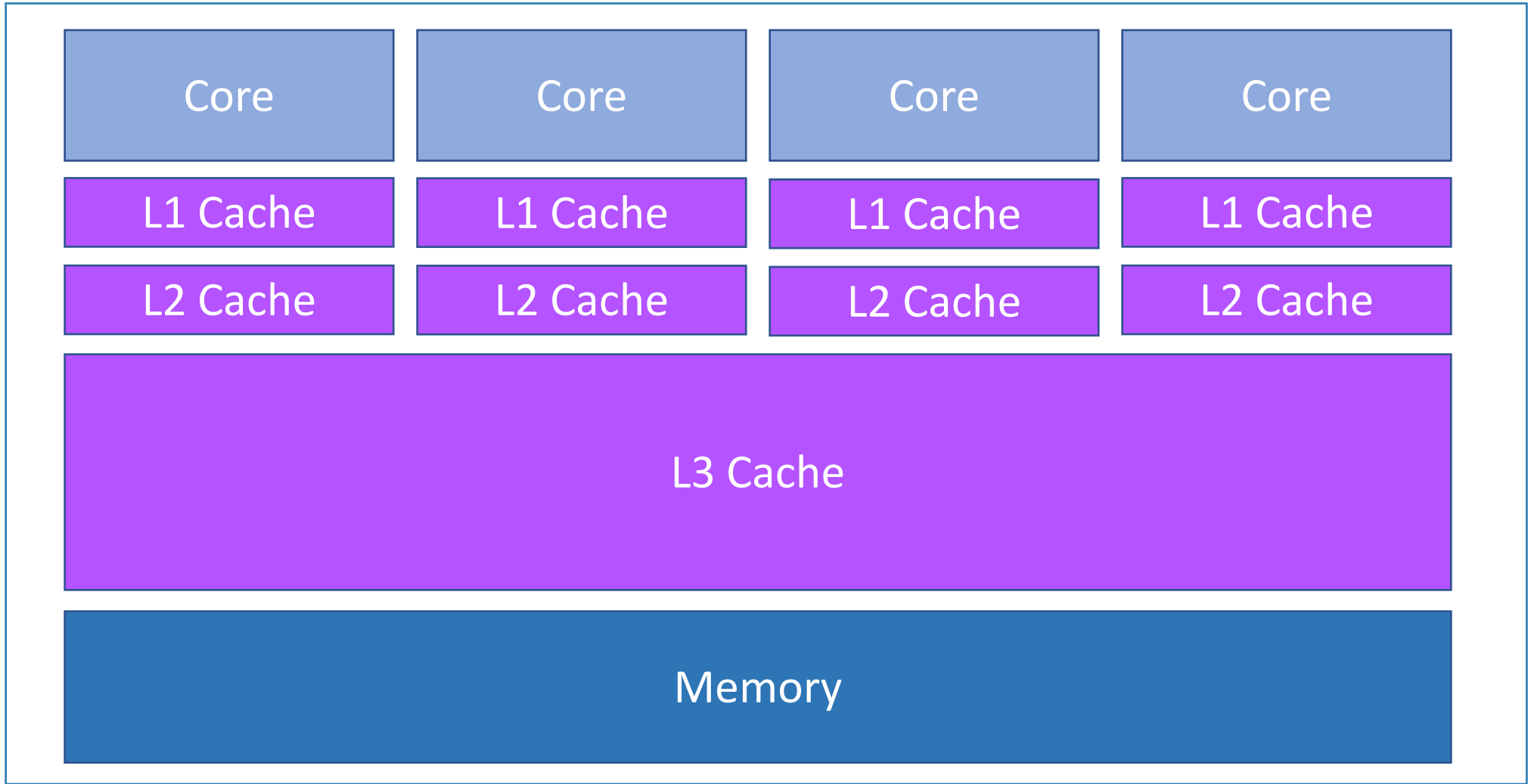
Multi-core CPU

- CPUs can have multiple cores, and cores can run in parallel



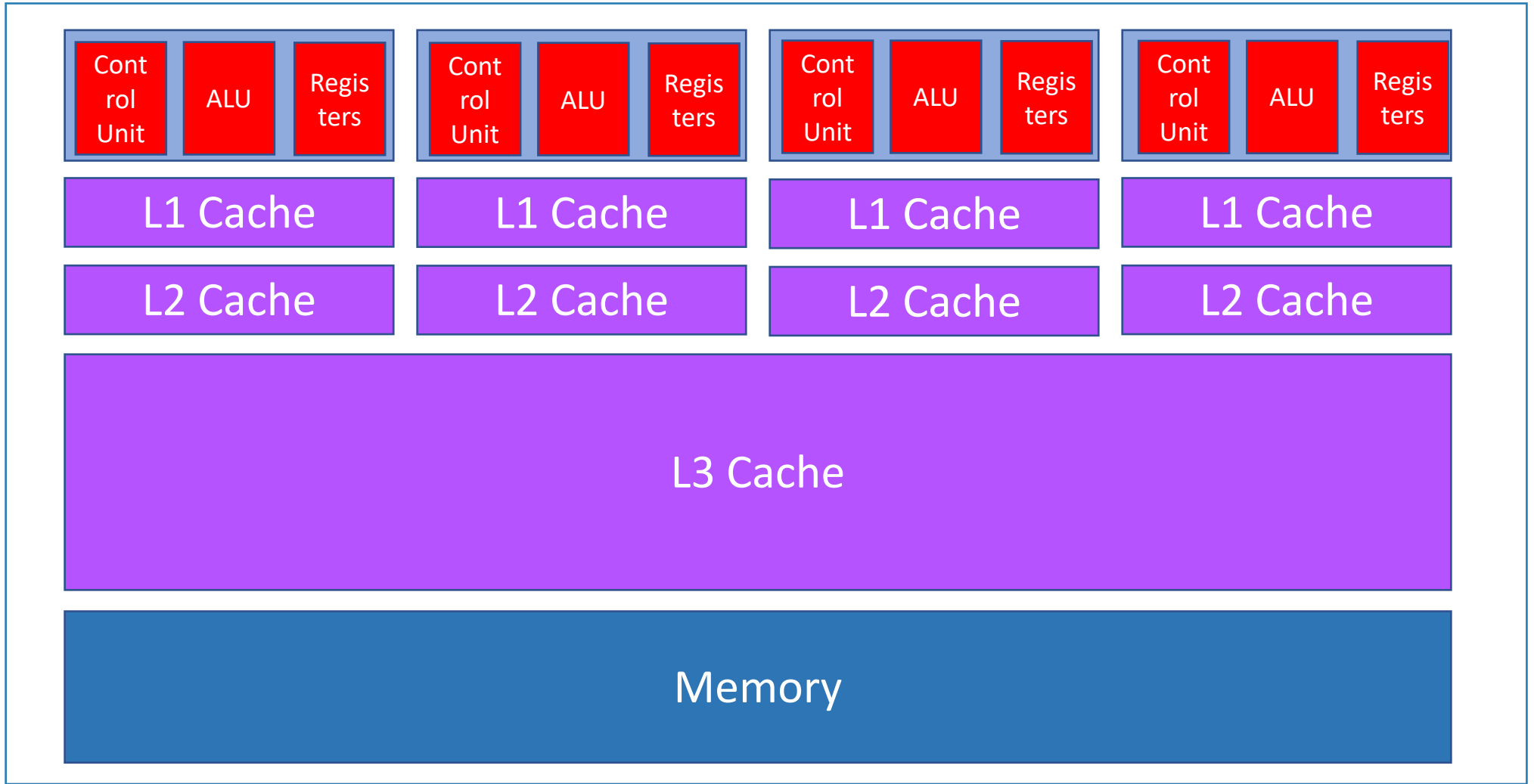
Multi-core CPU

- CPUs can have multiple cores, and cores can run in parallel



Multi-core CPU

- CPUs can have multiple cores, and cores can run in parallel

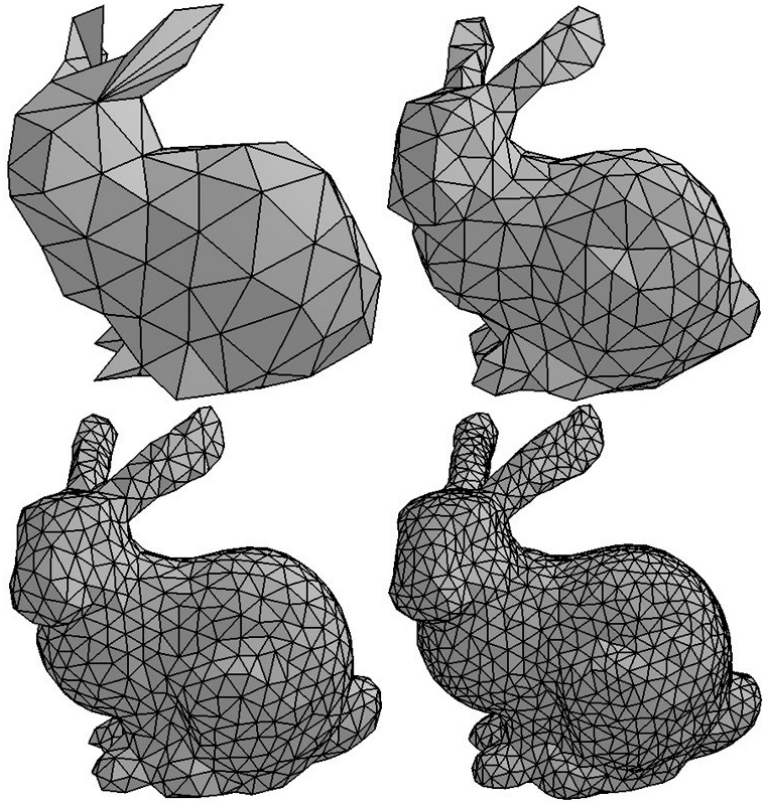


Gaming Applications

- Needed a lot of arithmetic for advanced fast rendering

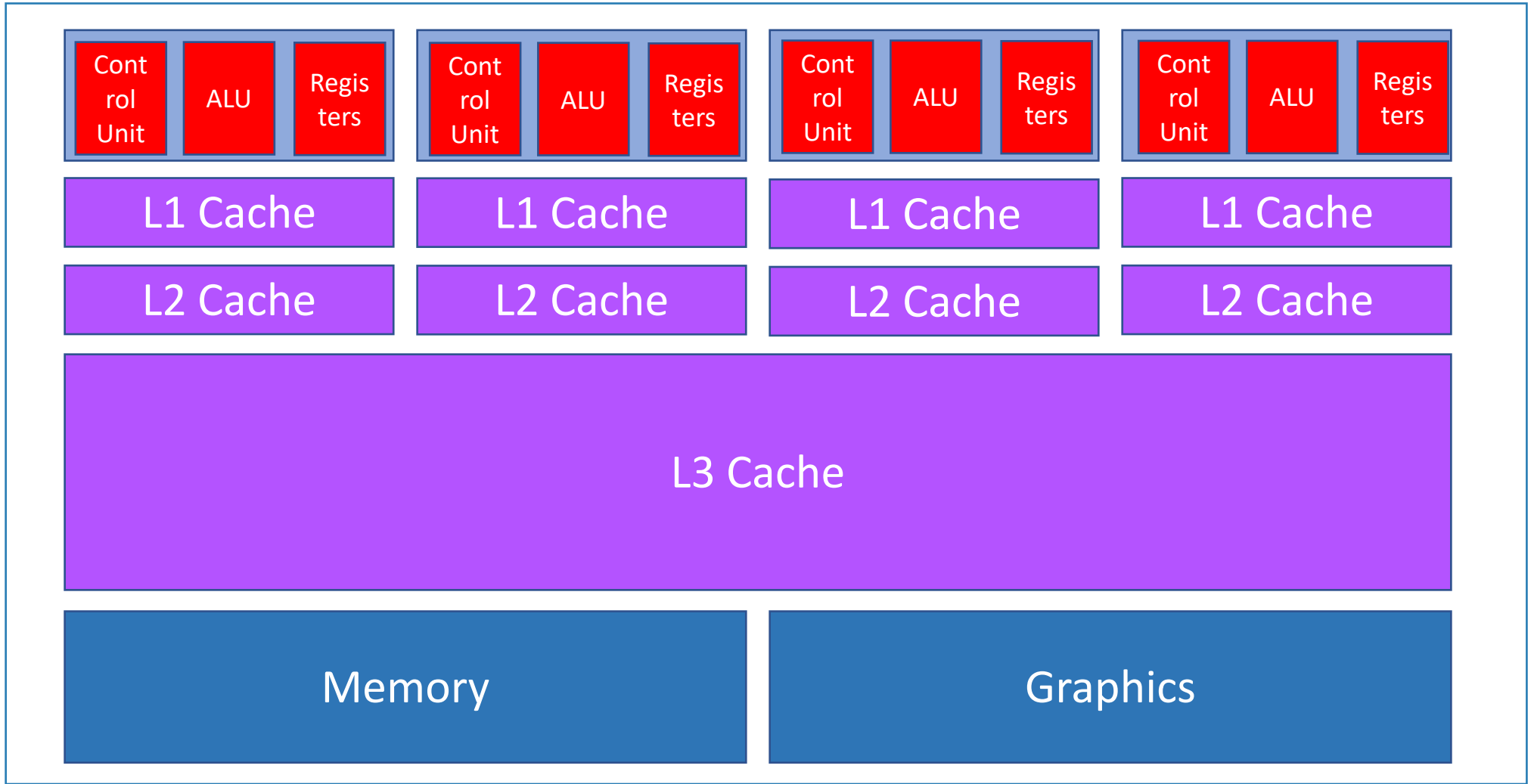
3D Object Rendering

- 3D objects broken up into polygons, usually triangles
- Coordinates stored as vectors
- Most of the mathematical operations involve geometry
- Moving, rotating, resizing, ray tracing, etc



Multi-core CPU

- CPUs started shipping with a graphics card

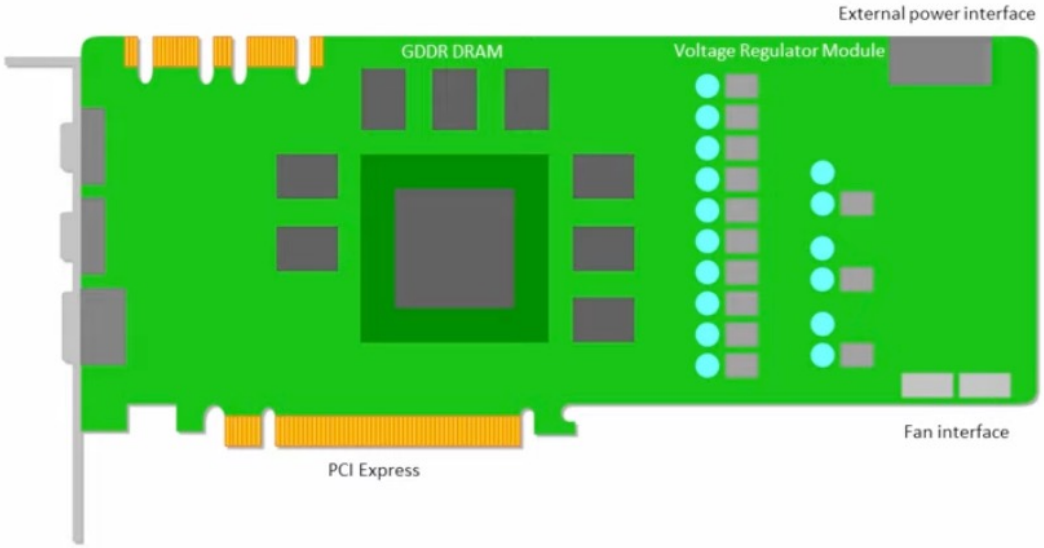


Graphics Card

- Specifically designed to handle graphics processing

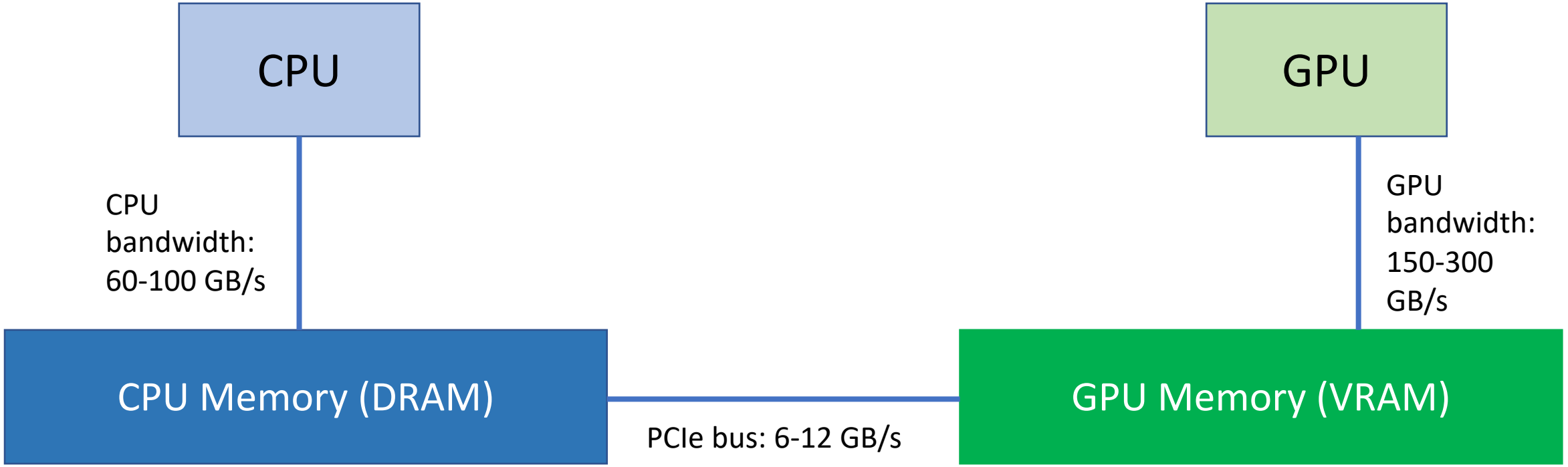
Graphics Card

- Has its own memory (VRAM), typically 8-10GB
- Has spot for cooling fan (can be noisy!)
- Often has external power supply (through CPU isn’t enough)



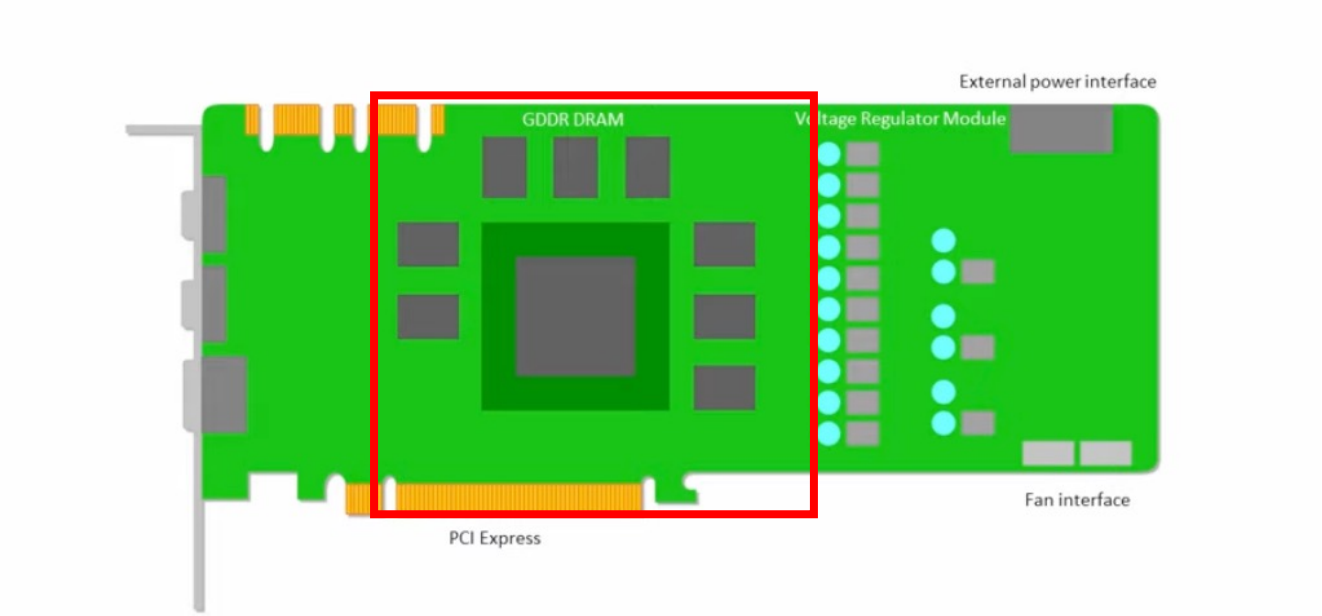
CPU Connects to GPU

- Must connect to CPU through PCIe bus
- Programmer must move data between DRAM and VRAM



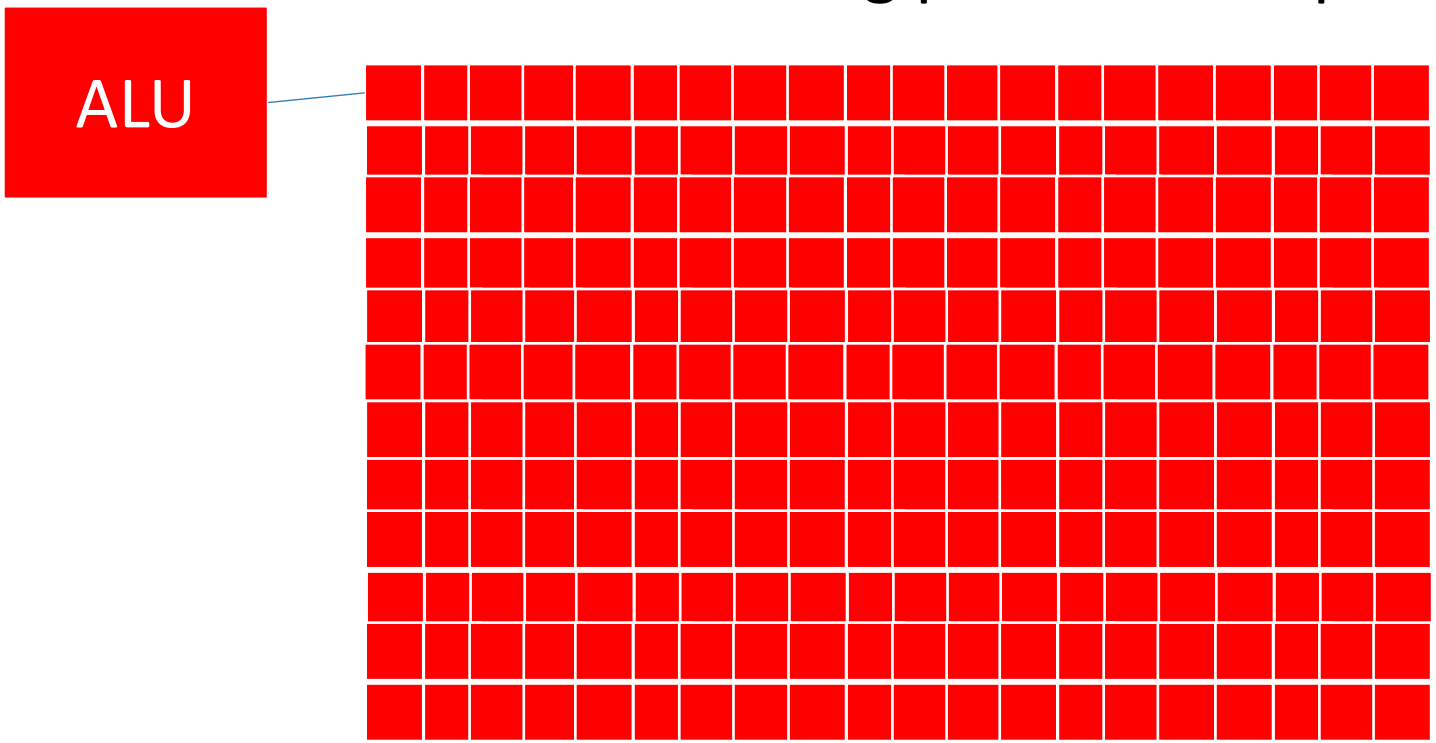
Graphics Card

- Most important part: Graphical Processing Unit
- Often the whole card is referred to as the GPU



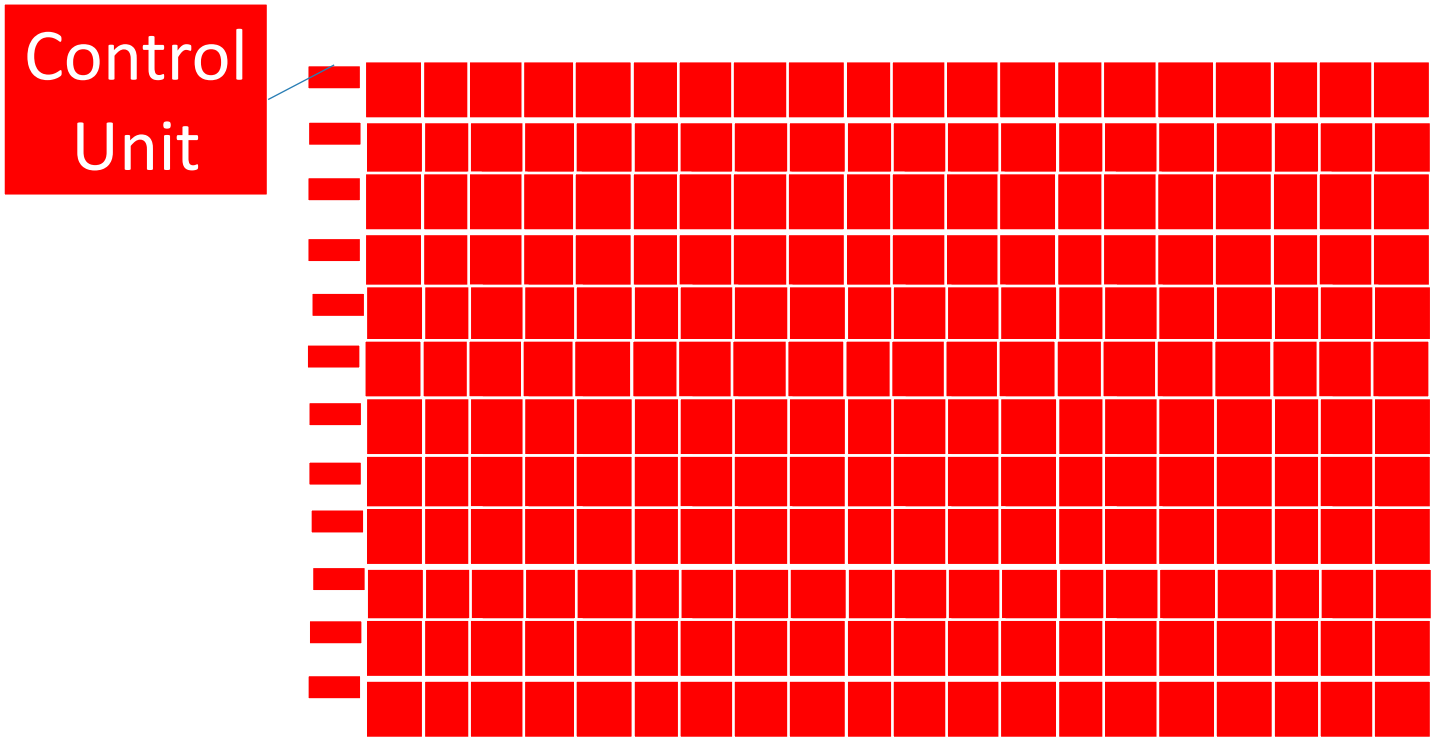
GPU Architecture

- Hundreds or thousands of ALUs, each working on its share of millions of data streams being processed in parallel



GPU Architecture

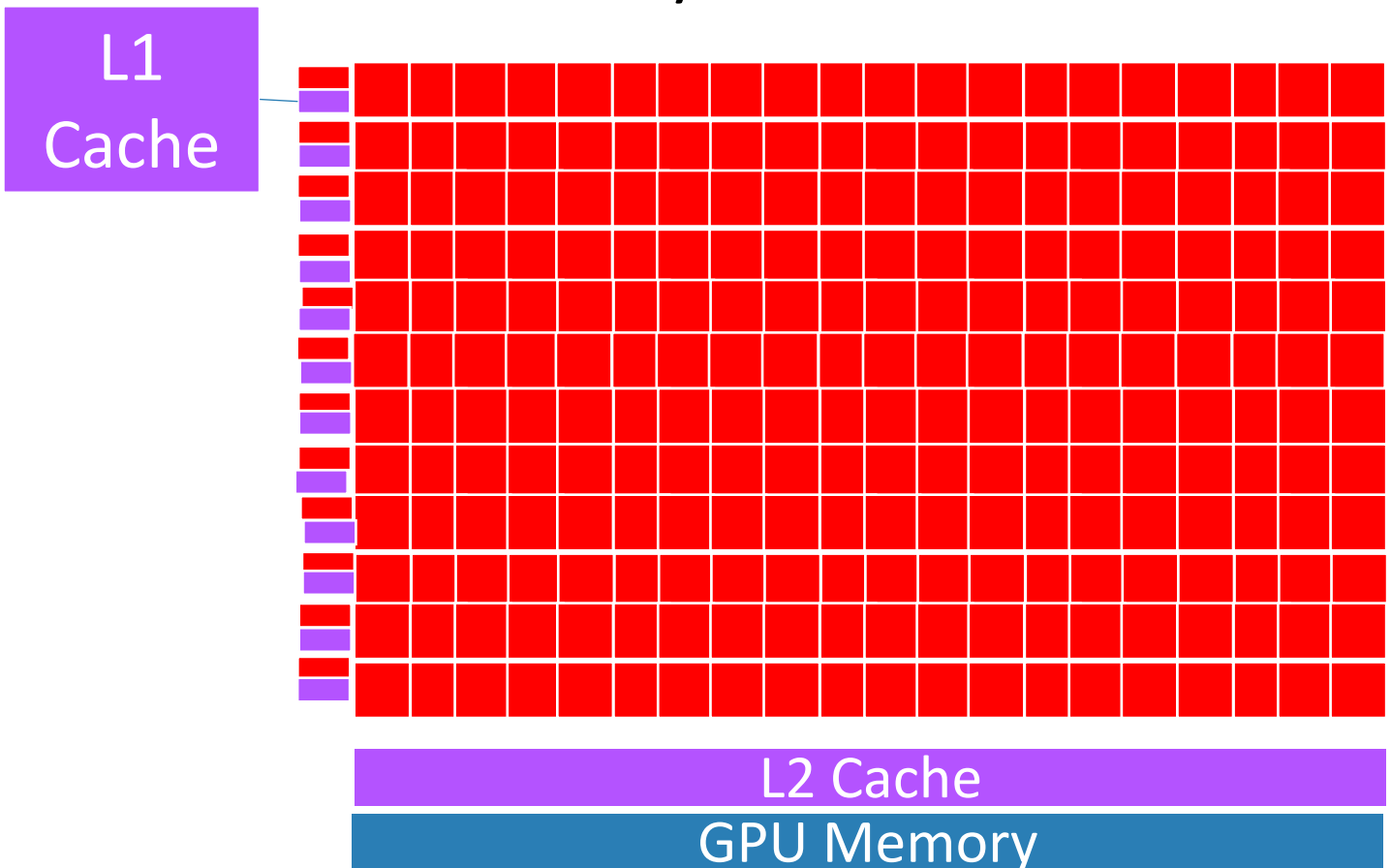
- Many ALUs execute the same instruction on different parts of data
- These can share a control unit



SIMD: single instruction, multiple data paradigm

GPU Architecture

- This group also shares cache, though there is also shared cache and shared memory across all ALUs



Why not always use GPU?

- CPU can handle many types of tasks
  - Spreadsheets, skype calls, music, etc
- GPU can do one thing very well

CPU

GPU

When should you use GPU?

- Because it has so many ALUs that can be coordinated to each take a subset of the data, GPU is very good for ”embarrassingly parallel” tasks
- GPU: lower latency, very high bandwidth

CPU

GPU

Next Time

- How do the ALUs in a GPU parallelize an embarrassingly parallel task like machine learning (aka matrix multiplication)?
- How do we as programmers: 1. Write code for the host and code for the device 2. Run device code from the host 3. Use device memory (transfer data between host and device)?