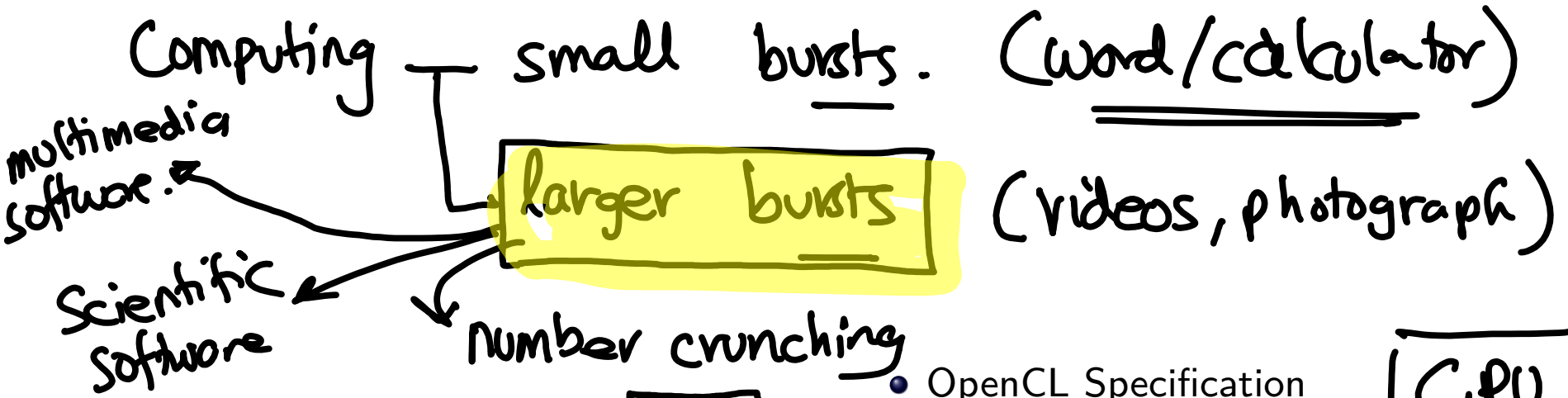
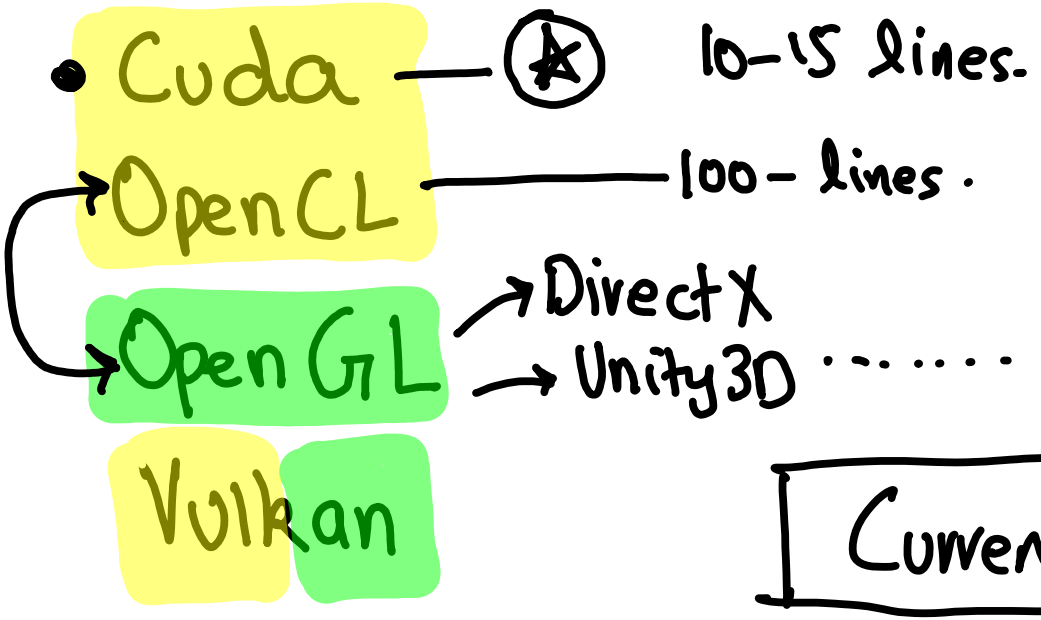
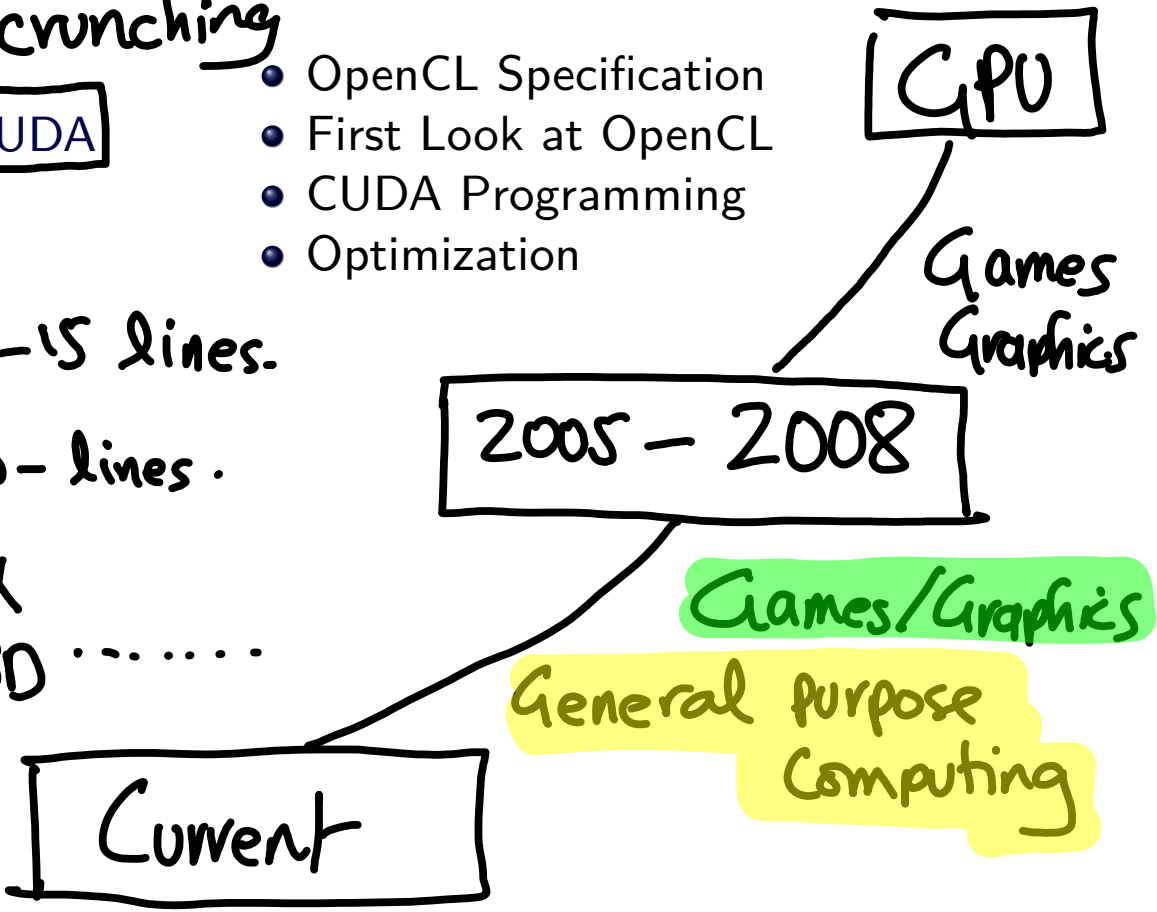
Syllabus

1. Vector Programming: OpenCL/CUDA
   - Overview
   - GPGPU's: OpenCL & CUDA
   - OpenCL Specification
   - First Look at OpenCL
   - CUDA Programming
   - Optimization







Vector Programming Overview

- Execution Models for GPU Architecture: MIMD, SIMD, SIMT

Figure 1: Fundamental Design Philosophy of CPU vs GPU

- **Control Logic**: Allow Parallel and/or Out-of-Order execution of threads. (Centralized on CPU, Decentralized on GPU)
- **ALU**: Perform arithmetic and bitwise operations (One ALU for each core on CPU, One ALU for each core on GPU, or One Arithmetic Unit (FPU) for each core)
- **Cache**: On-chip Memory to reduce instruction and data access latencies (Very small capacity on GPU)

Vector Programming Overview

- Execution Models for GPU Architecture: MIMD, SIMD, SIMT

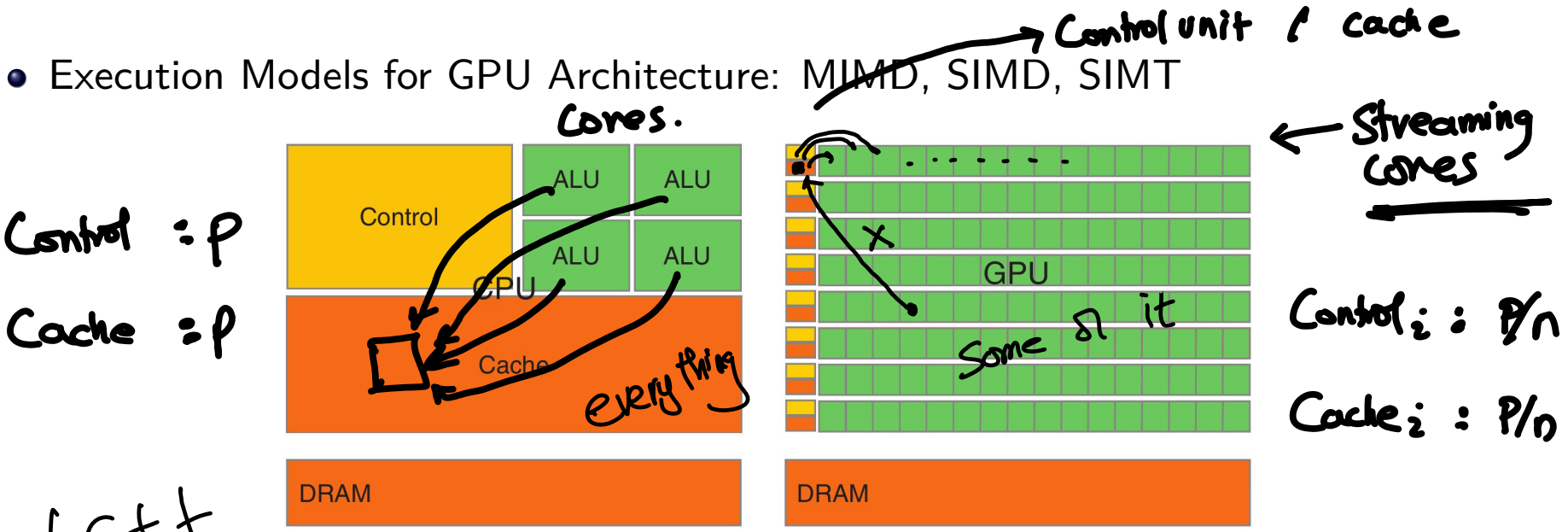


Figure 1: Fundamental Design Philosophy of CPU vs GPU

- **Control Logic**: Allow Parallel and/or Out-of-Order execution of threads. (Centralized on CPU, Decentralized on GPU)
- **ALU**: Perform arithmetic and bitwise operations (One ALU for each core on CPU, One ALU for each core on GPU, or One Arithmetic Unit (FPU) for each core)
- **Cache**: On-chip Memory to reduce instruction and data access latencies (Very small capacity on GPU)

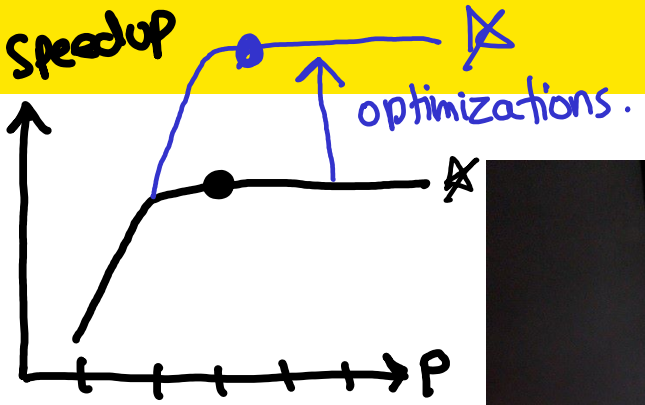
Vector Programming Overview (cont.)

- **DRAM CPU**: Off-chip Memory to store different processes

Why are people switching to GPU's?

- Performance Reasons (Offload numerically intensive parts to GPU)
- Processor availability in Market (Program for the dominant processor. 10 years ago, parallel parallel computing limited to governments and large corporations/universities. This has all changed now with GPUs, thanks to video games.)
- Massive scalability in limited space (Embedded applications requiring parallelism could not include large cluster-based machines. With GPUs, they can)
- IEEE Floating Point Compliancy (Early GPU's were not entirely IEEE compliant. Hence programmers refrained to use them. This is now almost history, unless you buy an old GPU)
- Graphics Programming no longer required to operate on Graphics Cards. We have **GPGPU** compliant API's.

Example: NVIDIA Kepler K40



Big Data

- GP-GPU, Scientific Computing
- Slave Processors
- GPU Giants (NVIDIA + AMD)
- CUDA: NVIDIA based GPU's
- OpenCL: Open coding standard for cross-device execution (Mobile Phones, GPU, CPU, Altera FPGA's), established by Khronos Group (2008)

-	Kepler K40	Intel i7-4900MQ
Cores	2,880	4 (8 Threads)
RAM	12 GB	
Cache	48 Kb	8 Mb
Clock	876 MHz	2.8 GHz
Bwidth	288 GB/s	25.6 GB/s
FLOP (d)	1.40 TFLOP	13.97 GFLOP
FLOP (f)	4.29 TFLOP	25.76 GFLOP

NVIDIA k40, 2880 cores, 12 GB RAM

2015 K40

2020 P40 ~4000

Latency Hiding

روی time 1/1 min

Kitchen — 1 — 1 minutes

1 Hz

10/1 minut

10 — 1 minutes

10Hz 6 second

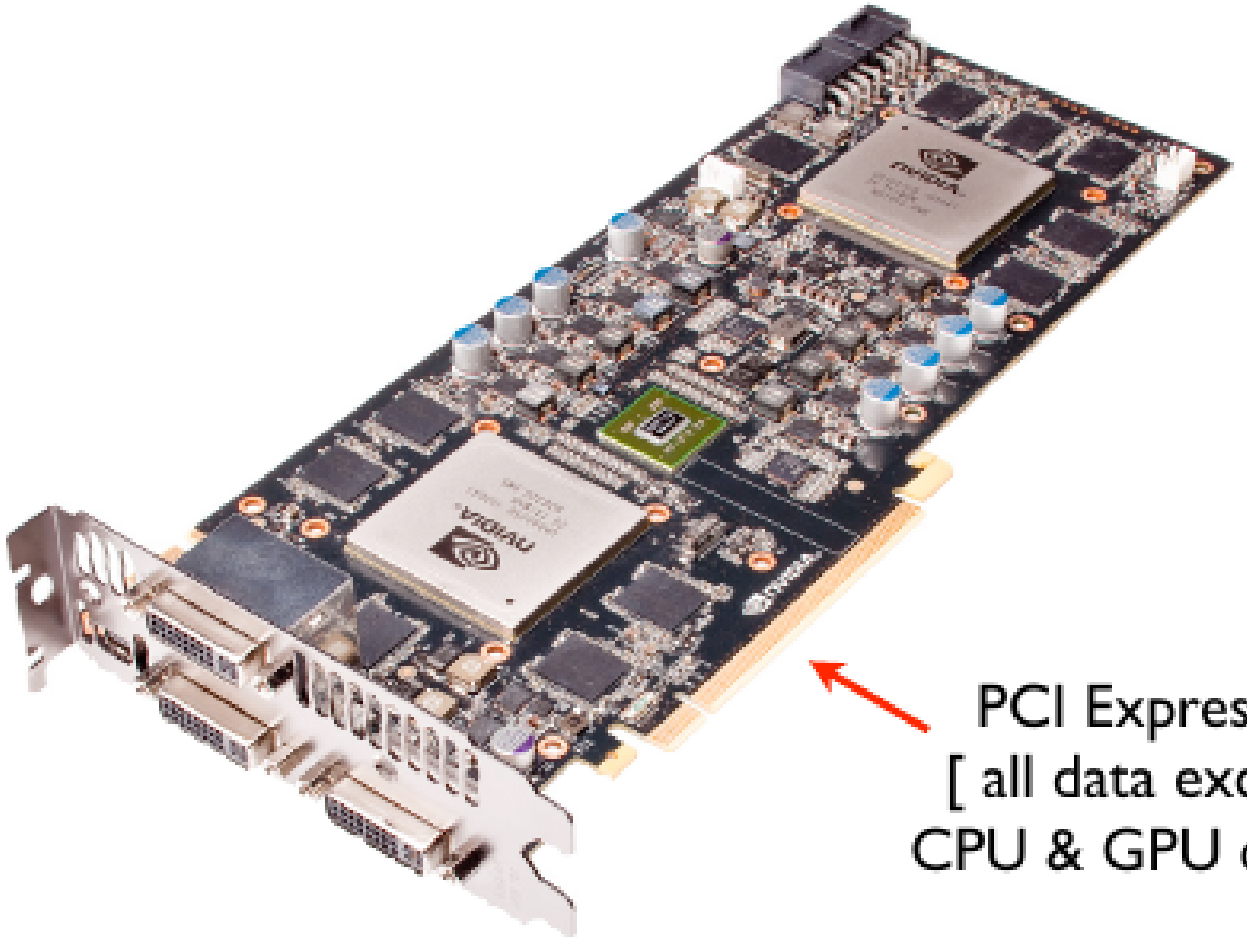
Example: NVIDIA GTX 590

GTX 590 GPU

GTX 590 GPU

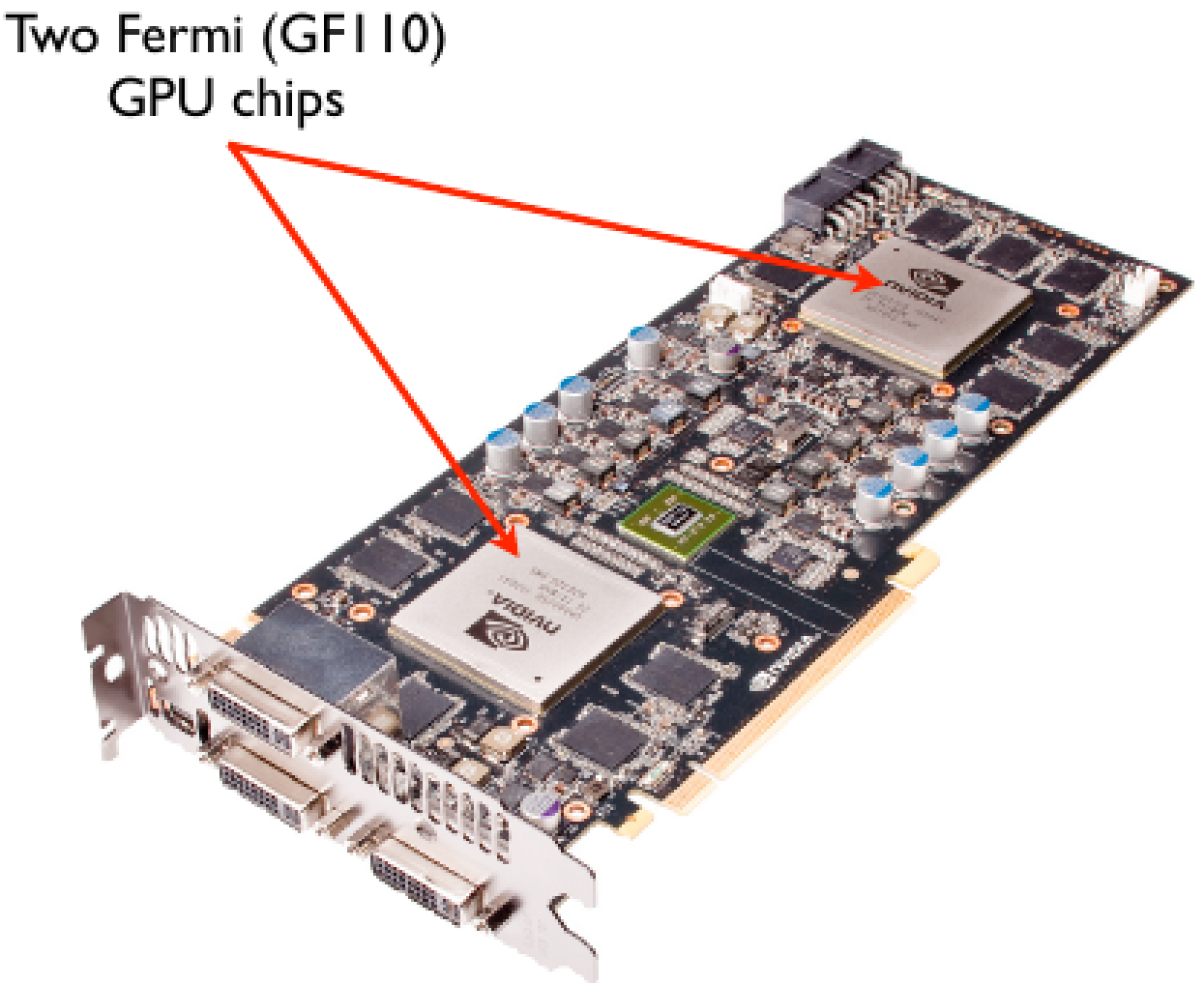
- Cores: 1024, Processor Clock: 1215 MHz, Memory: 3 GB

Example: NVIDIA GTX 590 (cont.)

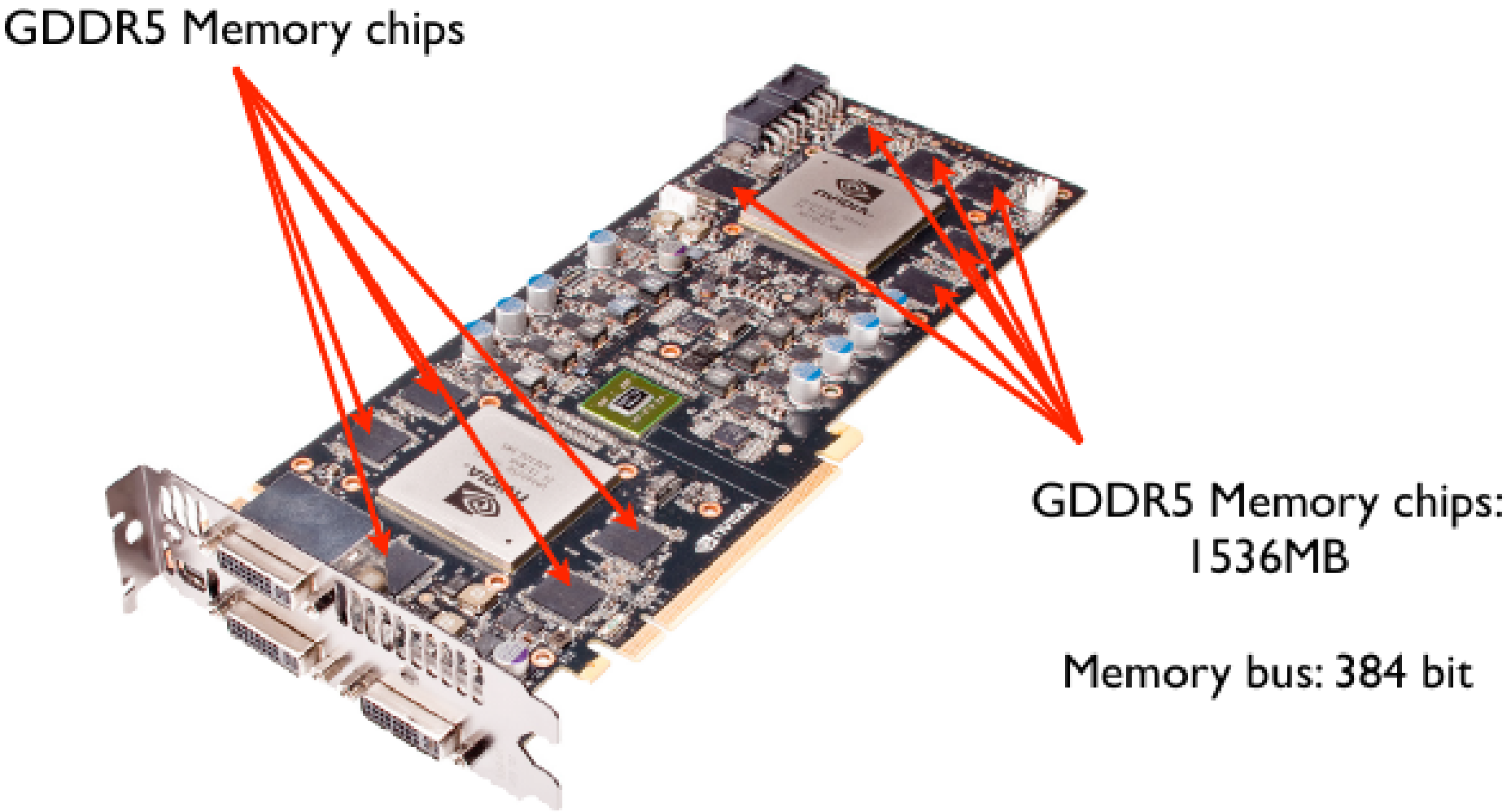


PCI Express 2.0 interface
[all data exchange between
CPU & GPU crosses this bus]

Example: NVIDIA GTX 590 (cont.)



Example: NVIDIA GTX 590 (cont.)



Example: NVIDIA GTX 590 (cont.)

Fermi: GPU <> GDDR5
memory bandwidth ~200 GB/s

PCI Express 2.0 bus
bandwidth ~6 GB/s

Xeon 5520 <> DDR3
memory transfer rate < 25 GB/s

Ethernet bandwidth:
~0.2 GB/s

USB3 bandwidth:
< 0.5 GB/s

SATA3 HD bandwidth:
< 0.8 GB/s

Figure 2: The Bottleneck

Trying it Out

Access Details

```
ssh 121.52.146.108 -l cdc-username -XY
```

where, **username** is your FAST-roll number
and **password** is same as username (one time only)
Example session with password = p116003 would be:

```
ssh 121.52.146.108 -l cdc-p116003 -XY
```

Introduction to GPGPU's

Open Computing Language (**OpenCL**)

- An **open** standard from **Khronos**; the makers of OpenGL (v1.0 Release December 2008)
- Cross Platform/Vendor/Architecture (CPU, GPU, DSP, FPGA, ...)
- GPU Giants (NVIDIA + AMD)
- Other Major Players (Apple, Intel, Qualcomm, Samsung, Xilinx, Altera)
- OpenCL is a {Standard, Language (based on C99), API/Library, Runtime SIMT based Compilation and Execution Environment}
- Two-way inter-operatable with OpenGL

Compute Unified Device Architecture (**CUDA**)

- **Proprietary** platform/API, released by **NVIDIA** (v1.0 released in January 2007)
- Handles only one platform/vendor, i.e., NVIDIA manufactured Graphic Cards
- CUDA is a {Language, API/Library, Non-runtime compilation environment, and Runtime execution environment}
- Inter-operatability with OpenGL is one-way (OpenGL can view CUDA buffers, but CUDA cannot view OpenGL buffers)

Introduction to GPGPU's (cont.)

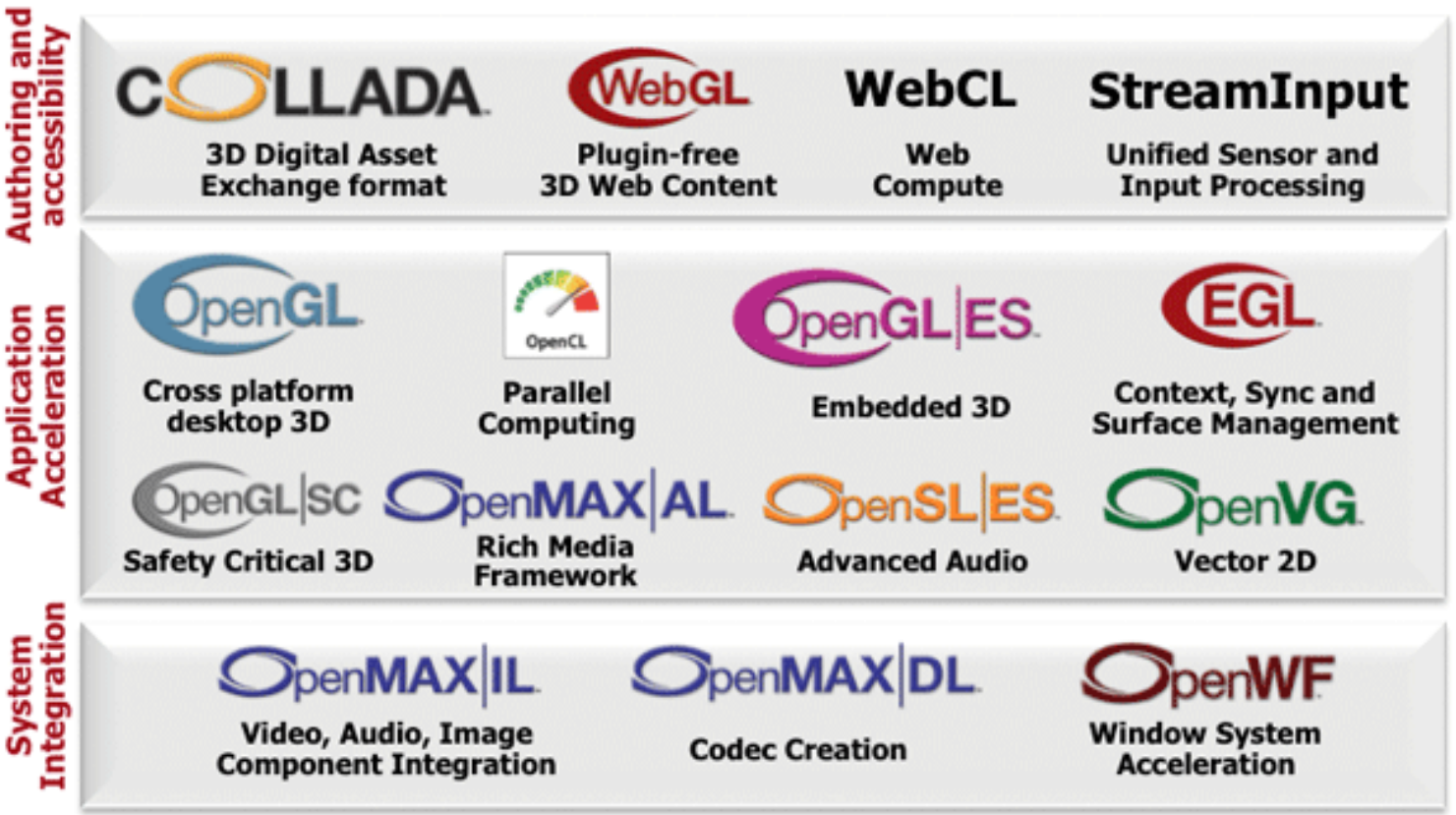


Figure 3: Khronos Group Open Specifications

[Img] http://www.khronos.org/about

Data Migration