High Energy Physics

# Traineeships in Advanced Computing for High Energy Physics (TAC-HEP)

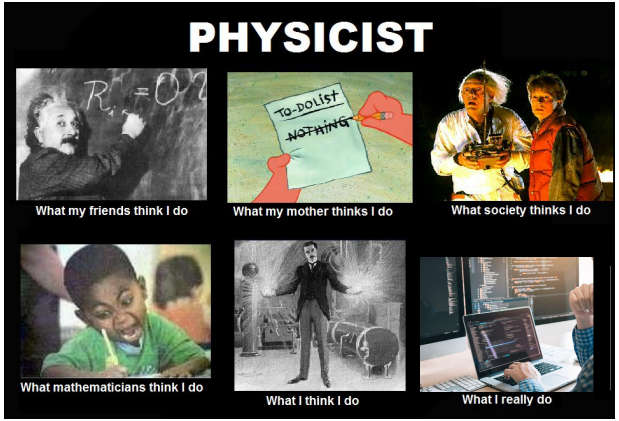
## GPU & FPGA module training

**Week 1** : Introduction to GPUs and heterogeneous computing

Lecture 1 - January 24th 2023

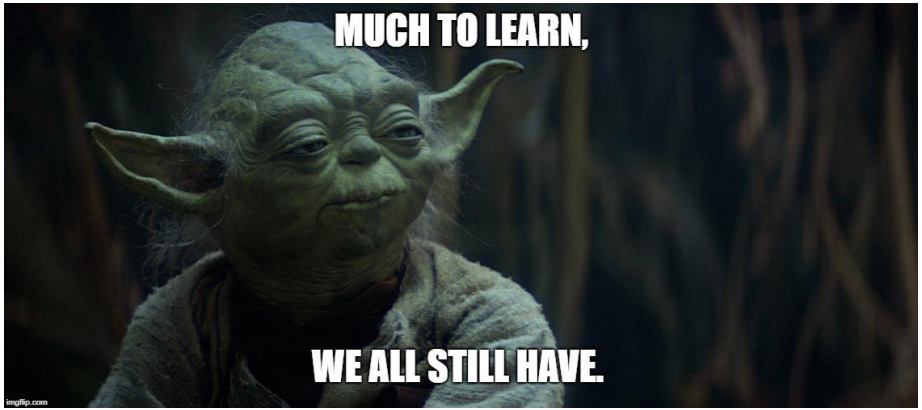
# Goal of this training module

- What does a daily life of an experimental physicist (usually) look like ?
  - Our detectors read collision or cosmic data and we want to reconstruct the physics quantities. What do we do ?
    - We write code
  - We want to check the quality / measure the performance of our reconstructed objects. How do we do that ?
    - We write code
  - We want to use the reconstructed data to perform our measurements or search for new physics. In order to do our statistical analysis / create histograms and figures:
    - Guess what, we write code!
- We usually learn how to code as we go
  - Why not try out some training?



# What we will (hopefully) learn in this training

- Get familiar with the concept of hardware accelerators and their applications
- Learn about heterogeneous computing
- Brush up some of our C++
- Become familiar with the CUDA programming model




- Write our first CUDA scripts
- Learn how to profile a piece of code and interpret the output
- Profile C++ code - identify bottlenecks and offload to GPU

# Overview of today's lecture

- The Central Processing Unit (CPU)
- Hardware accelerators: types and applications
- The Graphics Processing Unit (GPU)

# The CPU and hardware accelerators

# Central processing unit (CPU)

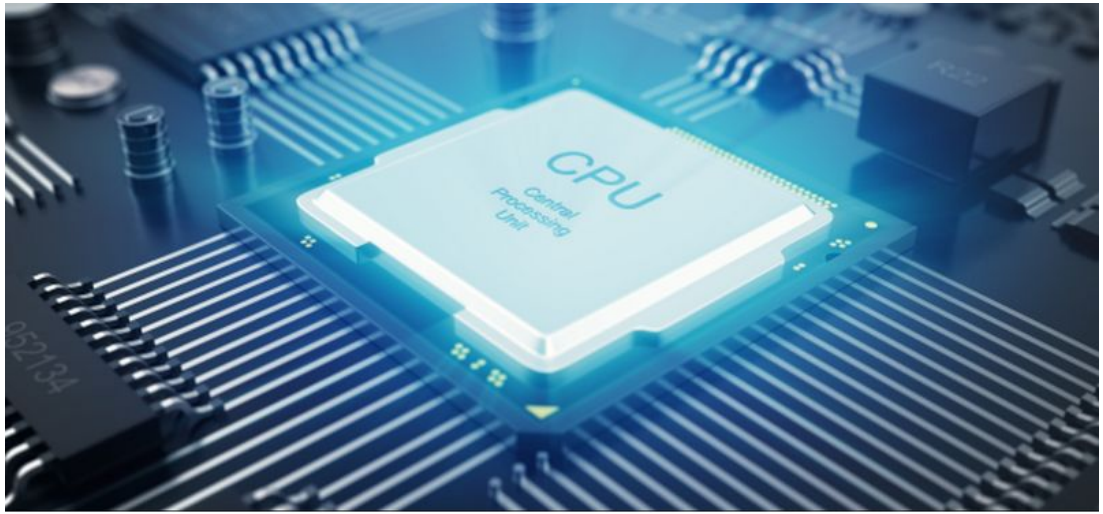
**Silicon-based micro-processor**

Used in most of our computers since it can handle a variety of tasks.

Performs certain types of operations **serially** :

- Arithmetic (+,*)
- Logical functions (AND, OR, NOT)
- Input/Output (I/O) operation

Is able to execute a sequence of instructions, which constitutes the “program”

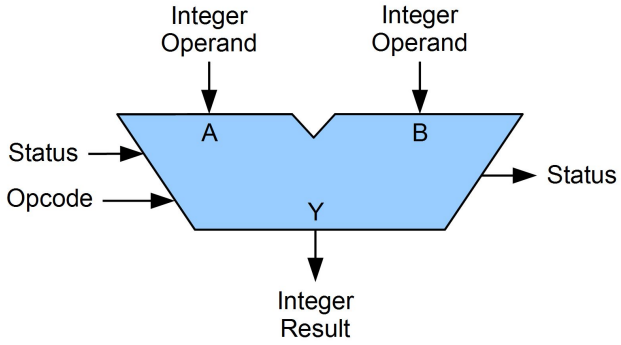



**The CPU is the brain of our computer, that reads information, performs calculations and moves it where it needs to go**

# How does a CPU work ? (1)

**Principal components of a CPU :**

- Arithmetic Logic Unit (ALU) :
  - Used to perform arithmetic and logic operations on integer binary numbers
- Processor registers :
  - A quickly accessible location available to a computer's processor
  - Is used to supply operands to the ALU and store the results of the ALU operations
- Control Unit (CU)
  - Is in charge of orchestrating fetching from memory / decoding / execution of instructions etc.




*** Schematic representation of an ALU**

* Image taken from [1]

# How does a CPU work ? (2)

CPUs are implemented on integrated circuit (IC) microprocessors :

- A single IC chip can have one or more CPU cores
- Microprocessor chips with multiple CPUs are **multi-core processors**
- Processor cores can also be multithreaded to create additional virtual CPUs

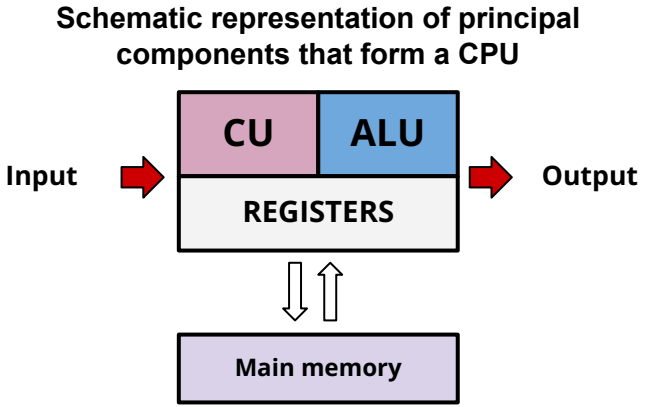
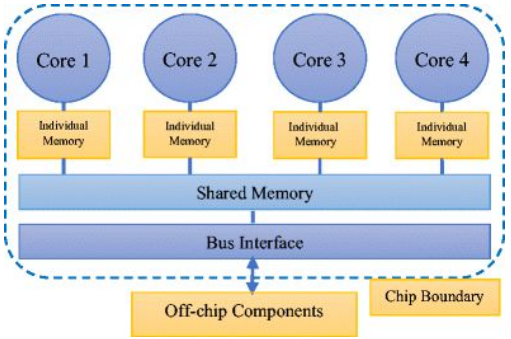



Image taken from [1]



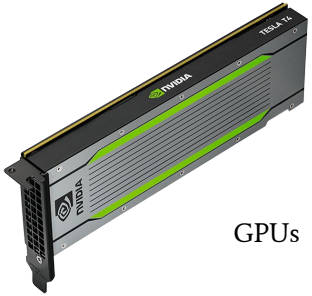
# Hardware accelerators

- Devices built for **executing specific tasks more efficiently** compared to running on the standard computing architecture of a CPU

- Part of our everyday lives :
  - Encryption, video stream decoding, 3D graphics acceleration, pattern/object recognition, machine learning, AI and many more

# Some types of hardware accelerators (1)

- **GPU** (Graphic Processing Unit)
  - Initially developed for graphics processing
  - Optimized for parallel processing of floating point operations & used in a variety of tasks



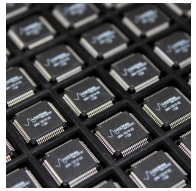
GPUs

- **FPGA** (Field-Programmable Gate Array)
  - Integrated circuit (IC) configurable by the user and provides interface flflexibility
  - FPGAs can be reprogrammed to suit the needs of the application or required functionality


FPGAs

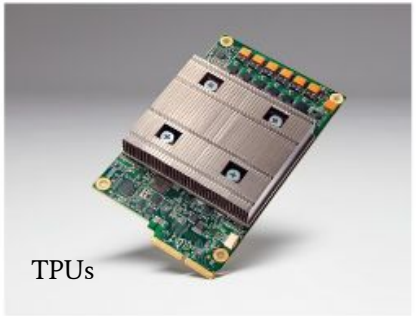
# Some types of hardware accelerators (2)

- **ASIC** (Application-Specific Integrated Circuit)
  - IC chip customized for a particular use
  - i.e. lower precision and/or optimised memory usage to maximize throughput



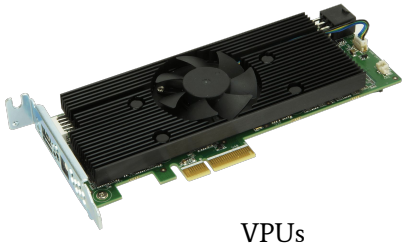
ASIC

- **TPU** (Tensor Processing Unit)
  - Optimised to perform matrix-multiplication operations / used in i.e. NN and RF training



TPUs

- **VPU** (Vision Processing Unit)
  - Used to accelerate machine vision algorithms, i.e. CNNs , AI etc.



VPUs

# Multi-core vs many-core architectures

**Multi-core processors**

- Built on a single IC with two or more processing units (**cores**)
- Emphasis on high single-thread performance
- Better latency
- Can be complemented by a many-core system

**Many-core processors**

- Much higher degree of parallelism compared to a many-core processors
- Emphasis on maximizing throughput
- Lower single-threaded performance and worse latency compared to multi-core processors

# Flynns classification of computer architecture

- Based on the number of instruction and data streams that can be processed simultaneously, computing systems are classified into four major categories

**i.e. single cpu-core machine**

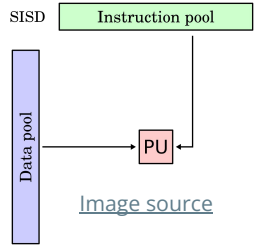



Image source

**i.e pipeline architectures - not commonly used**

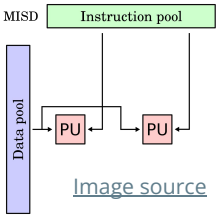



Image source

| | |
|---|---|
| **SISD** (**S**ingle **I**nstruction **S**ingle **D**ata) | **MISD** (**M**any **I**nstructions **S**ingle **D**ata) |
| **SIMD** (**S**ingle **I**nstruction **M**any **D**ata) | **MIMD** (**M**any **I**nstructions **M**any **D**ata) |

**Data streams**

**Instruction streams**

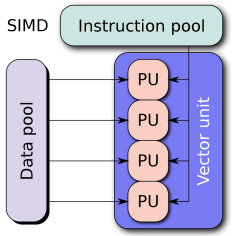



Image source

**i.e. vector processor cpus**

**i.e multi cpu-core machines / grid computing etc.**

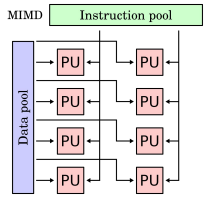



Image source

# SIMD vs SIMT

- **Single instruction, multiple threads (SIMT)** is an execution model which combines the SIMD model and multithreading.
  - **The GPU computing paradigm follows the SIMT approach**
- SIMD and SIMT approaches though similar have some differences :

**SIMD**

- Uses vectors
- Instructions executed in lockstep
- No synchronization required

**SIMT**

- Uses threads
- Not all threads are processed in lockstep
- Synchronization is required

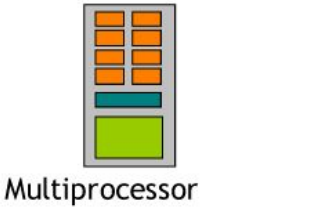
# The Graphic Processing Unit (GPU)

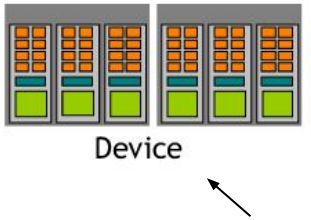
**GPUs are similar to CPUs :**

- Silicon based micro-processor that contain cores, registers, memory, and other components.

**But also very different :**

- **Many-core processor**
- Follows the **Single instruction, multiple threads (SIMT)** execution model
- GPU acceleration emphasizes on :
  - **High Data Throughput and Massive Parallel Computing:** a GPU consist of hundreds of cores performing the same operation on multiple data items in parallel.


In the CUDA terminology the **GPU** is referred to as the **"device"**

# Wrapping-up

# Overview of today's lecture

- Hardware accelerators are used in combination with CPUs to executing specific tasks more efficiently

- There are many types of hardware accelerators, both general purpose as well as manufactured targeting specific applications

- The GPU is a **many-core processor** that follows the **multiple threads (SIMT) execution model**
    - It has thousands of cores that can provide massive parallelization

# Next time

- We will learn about :
  - Differences between the GPU and the CPU
  - Differences between the GPU and the FPGA
  - Heterogeneous computing
  - The computing challenges in HEP
  - GPU applications in HEP

# BACK-UP