Programming for GPU

CUDA

CUDA − **C**ompute **U**nified **D**evice **A**rchitecture.

It is an extension of C programming, an API model for parallel computing created by Nvidia.

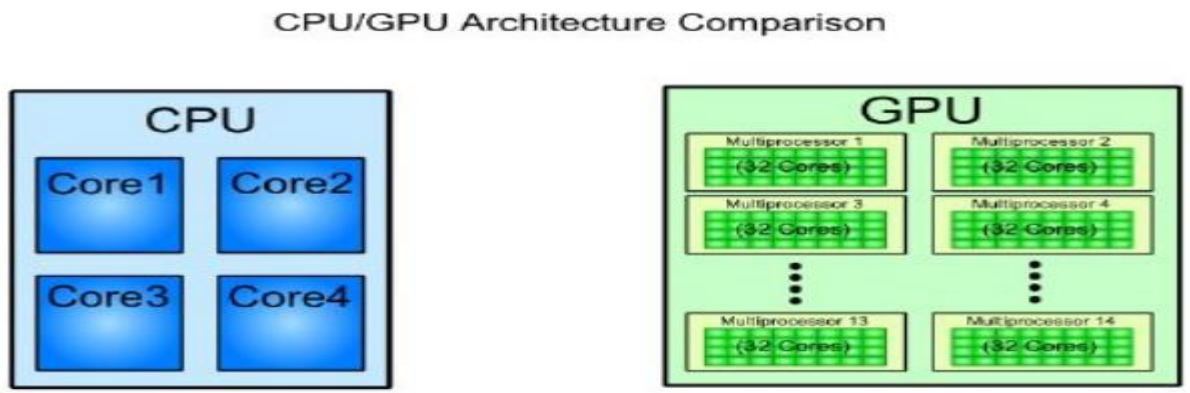
Programs written using CUDA harness the power of GPU. Thus, increasing the computing performance.

Parallelism in the CPU

1. Pipelining
2. Superscalar
3. SMT (Simultaneous Multithreading).

Introduction to the GPU:

1. Graphics Processing Units (GPUs) .
2. A GPU comprises many cores (that almost double each passing year), and each core runs at a clock speed significantly slower than a CPU's clock.
3. GPUs focus on execution throughput of massively-parallel programs.
4. For example, the Nvidia GeForce GTX 280 GPU has 240 cores, each of which is a heavily multithreaded, in-order, single-instruction issue processor (SIMD − single instruction, multiple-data) that shares its control and instruction cache with seven other cores.



Difference Between GPU and CPU

Feature	GPU	CPU
Primary Function	Graphics rendering, parallel processing	General-purpose computing, sequential tasks
Architecture	Highly parallel architecture with thousands of cores	Fewer, more powerful cores optimized for sequential processing.
Parallel Processing	Suited for parallel tasks due to numerous cores.	Primarily designed for sequential processing tasks
Instruction Set	Limited instruction set, optimized for specific tasks (eg, matrix operations)	Comprehensive instruction set for a wide range of tasks.
Memory Hierarchy	Multiple levels of memory (global, shared, registers) for efficient parallel processing.	Typically has a cache hierarchy (L1, L2, L3 caches) optimized for sequential tasks.

Difference Between GPU and CPU

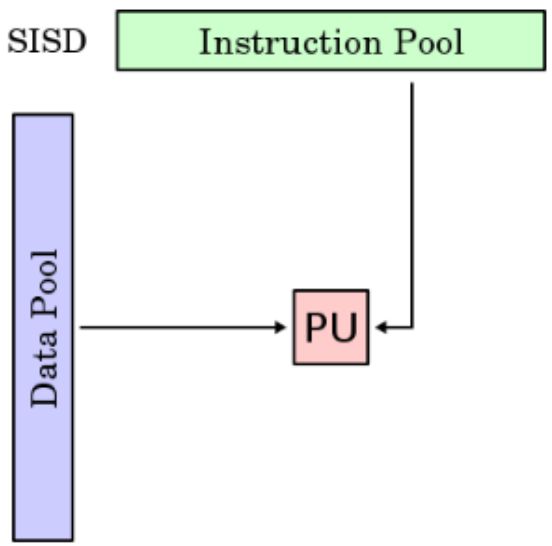
Feature	GPU	CPU
Clock Speed	Lower base clock speeds but compensated by adequate number of cores.	Higher base clock speeds optimized for sequential tasks.
Performance per Core	Lower performance per core due to specialization in parallel tasks	Higher performance per core for sequential tasks
Flexibility	Specialized for graphics and parallel computation, less flexible for general computing.	Versatile and flexible for various computing tasks
Energy Efficiency	Less energy efficient due to the emphasis on parallel processing tasks	More energy efficient for sequential tasks
Usage Scenarios	Ideal for workloads that support parallel processing, like gaming, AI, scientific simulations, etc.	Suited for general computing tasks such as running operating systems, office applications, etc.

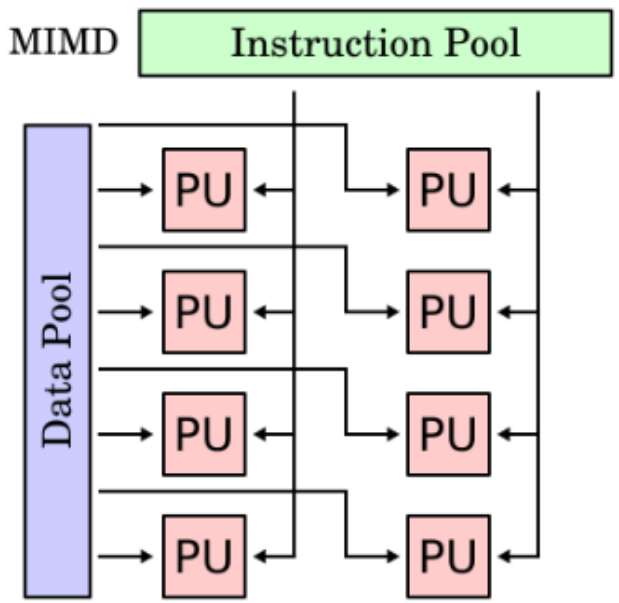
Difference Between GPU and CPU

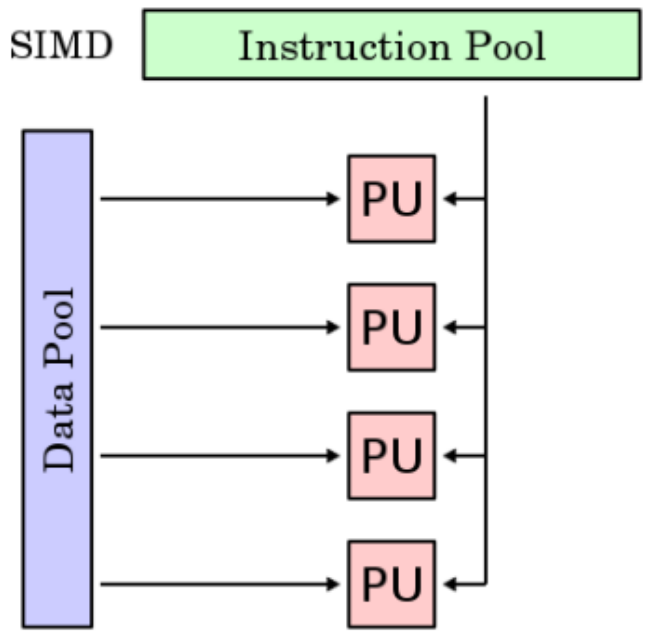
Feature	GPU	CPU
Cost	Relatively costlier, due to specialized hardware	Generally, more cost effective for general purpose computing
Programming Model	Requires specialized programming (e.g., CUDA, OpenCL) for optimal utilization.	Relatively standard programming languages (e.g., C, C++, Java) can be used.

SISD vs. SIMD

SISD	MIMD	SIMD
Single Instruction Single Data	Multiple Instruction Multiple Data	Single Instruction Multiple Data
Uniprocessor machines	Multi-core, grid-, cloud-computing	e.g. vector processors

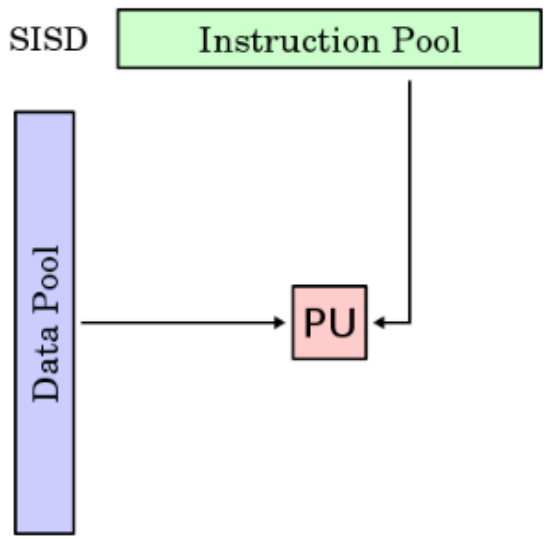


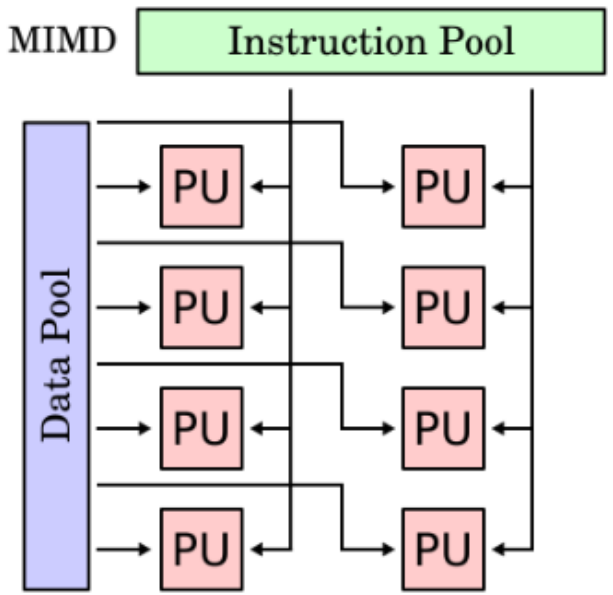


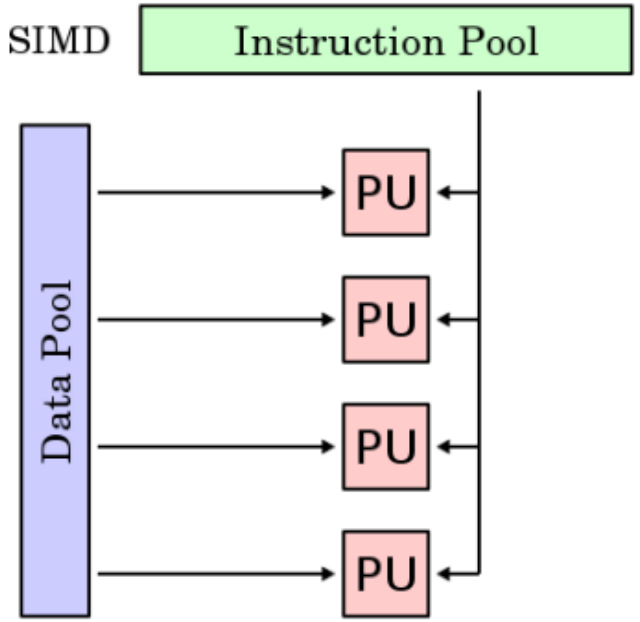


SISD vs. SIMD

SISD	MIMD	SIMT
Single Instruction Single Data	Multiple Instruction Multiple Data	Single Instruction Multiple Threads
Uniprocessor machines	Multi-core, grid-, cloud-computing	GPUs







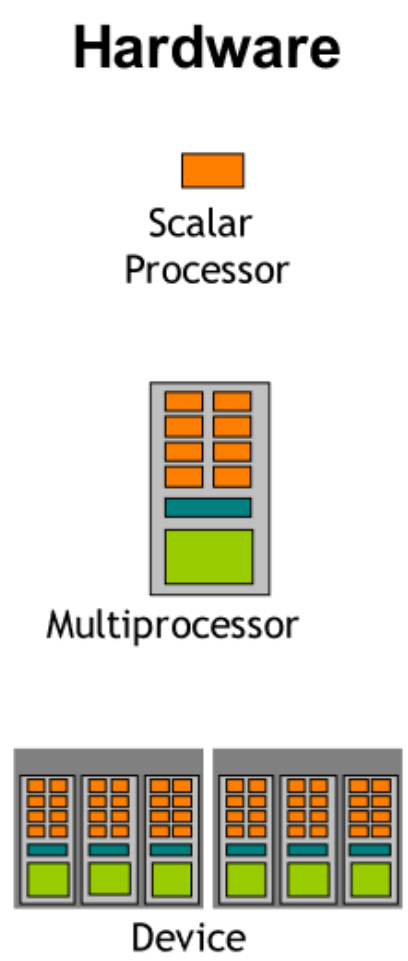
Single Instruction Multiple Threads

SIMT

- Similar to programming a vector processor
- Use threads instead of vectors
- No need to read data into vector register
- Only one instruction decoder available
  → all threads have to execute the same instruction

- Abstraction of vectorization:
  - Each element of vector is processed by an independent thread
  - One instruction fills entire vector
  - # of threads = vector size

What is a GPU?



(CUDA terminology)

- Several processors are grouped into a “multiprocessor”
- Several multiprocessors make up a GPU

NVIDIA GPU Architecture

- A scalable array of multithreaded Streaming Multiprocessors (SMs), each SM consists of
  - 8 Scalar Processor (SP) cores
  - 2 special function units for transcendentals
  - A multithreaded instruction unit
  - On-chip shared memory
- GDDR3 SDRAM
- PCIe interface

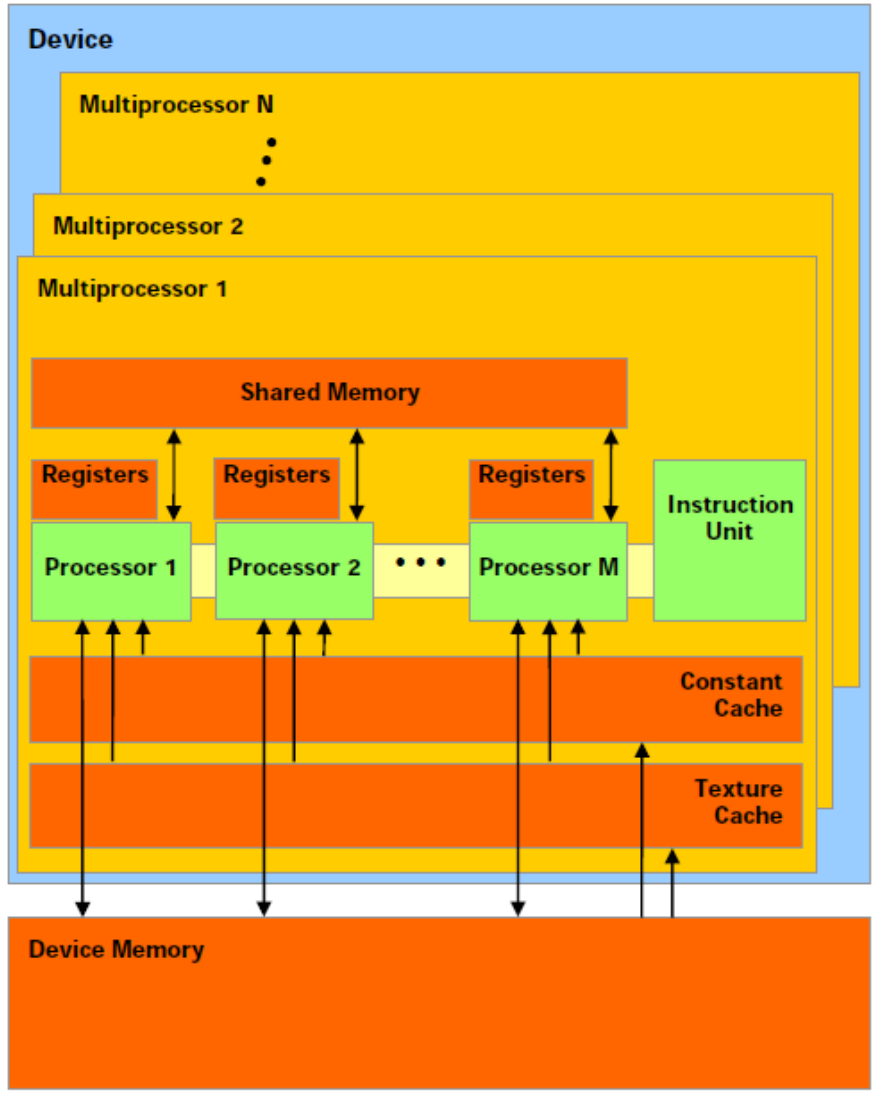
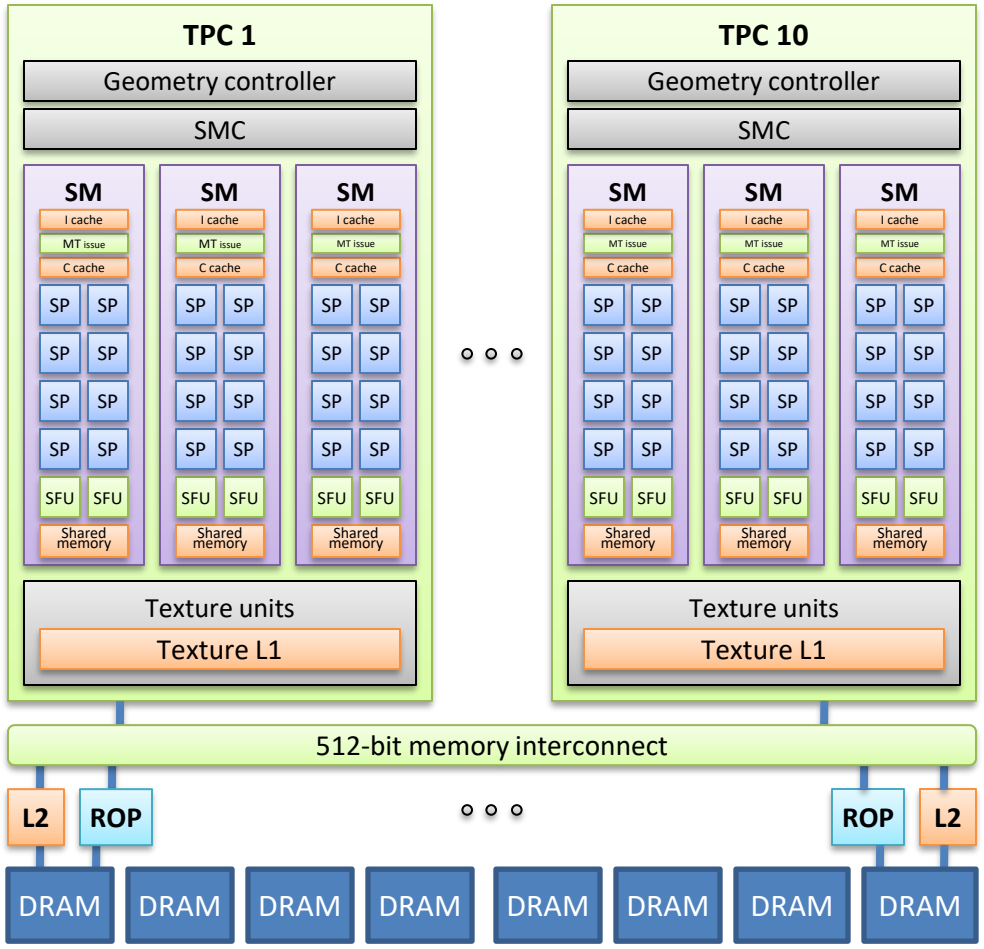


Figure is courtesy of NVIDIA

NVIDIA Tesla C1060 GPU



- 240 streaming processors arranged as 30 streaming multiprocessors
- At 1.3 GHz this provides
  - 1 TFLOPS SP
  - 86.4 GFLOPS DP
- 512-bit interface to off-chip GDDR3 memory
  - 102 GB/s bandwidth