Basics of CUDA

GROUP 1

KARTHIK SRIDHAR-B190467CS

PAVITHRA RAJAN- B190632CS

CLIFORD JOSHY- B190539CS

JESVIN SEBASTIAN MADONA-B190700CS

What is CUDA?

COMPUTE

UNIFIED

DEVICE

ARCHITECTURE

PURPOSE

- General purpose parallel computing platform for NVIDIA GPUs
- Solves complex computational problems

METHOD

- Leverages parallel computing powers of GPUs over CPU
- Task -> Several independent threads
- Acts as a SW layer- direct access to instruction set of GPU

GPU over CPU

CPU

- Generalist component-handles the main processing functions of a computer
- Lesser number of cores (2-64)
- Runs processes serially
- Better at processing one big task at a time.
- Minimizes latency

GPU

- Specialized component-handles graphic and video rendering
- Larger number of cores (thousands)
- Runs processes in parallel
- Better at processing several smaller tasks simultaniously
- Maximizes throughput

C / C++ EXTENSION

- CUDA C++ extends C++ by allowing the programmer to define C++ functions-Kernels

- When the kernels are called, they are executed N times in parallel by N different CUDA threads, as opposed to only once like regular C++ functions

Host : The CPU and its memory
Device : The GPU and its memory

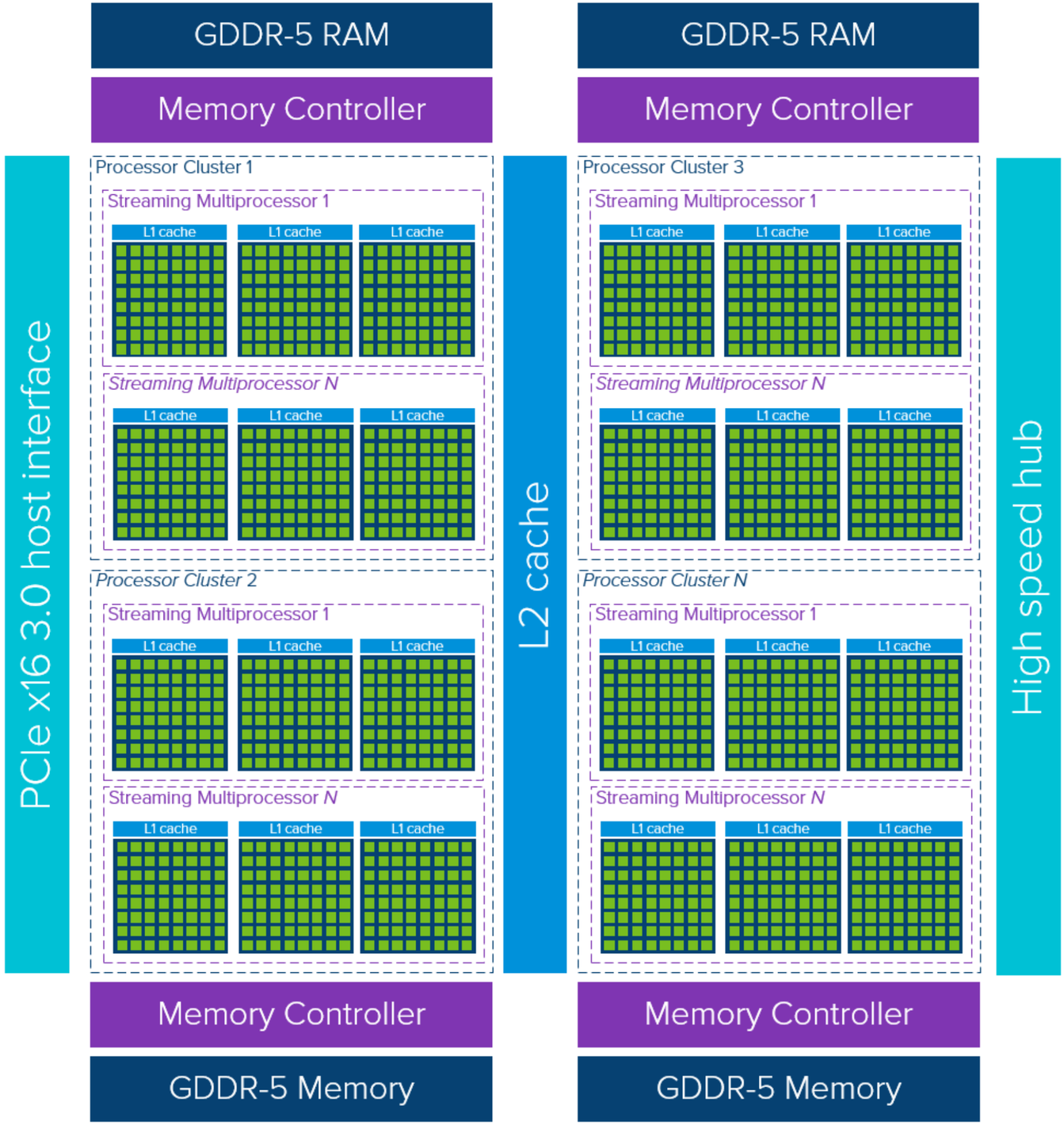
Sequence of Operations

Declare and Allocate → **Initialize** → **Transfer** → **Execute** → **Transfer**

- **Declare and Allocate**: Declare and allocate host and device memory
- **Initialize**: Initialize host data
- **Transfer**: Transfer data from the host to the device
- **Execute**: Execute one or more kernels.
- **Transfer**: Transfer results from the device to the host

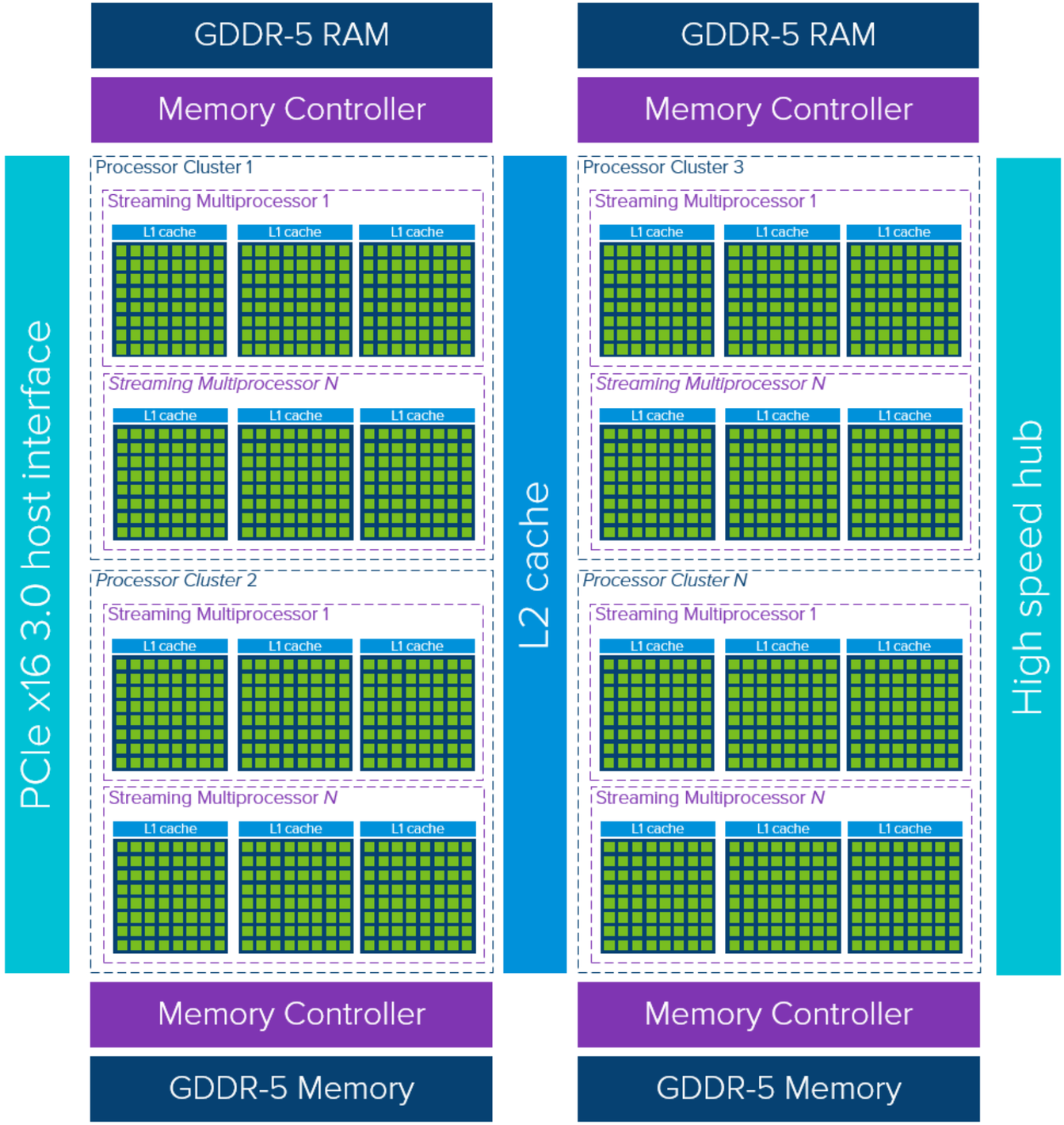
GPU Architecture-1

- A single GPU device consists of multiple Processor Clusters (PC) that contain multiple Streaming Multiprocessors (SM).
- Each SM accommodates an L1 instruction cache layer with its associated cores.
- Each SM uses a dedicated L1 cache and a shared L2 cache before pulling data from global GDDR-5/GDDR-6 memory.



GPU Architecture-2

- Compared to a CPU, a GPU works with fewer, and relatively small, memory cache layers.
- Since a GPU has more transistors dedicated to computation, it does not stress over memory access time.
- The potential memory access latency is masked as long as the GPU has enough computations at hand, keeping it busy.



7 Components of GPU Architecture

GCA - GRAPHICS AND COMPUTE ARRAY

GMC - GRAPHICS MEMORY CONTROLLER

VGA BIOS - VIDEO GRAPHICS ARRAY BASIC INPUT/OUTPUT SYSTEM

BIF - BUS INTERFACE

PMU - POWER MANAGEMENT UNIT

VPU- VIDEO PROCESSING UNIT

DIF- DISPLAY INTERFACE

GPU ARCHITECTURE

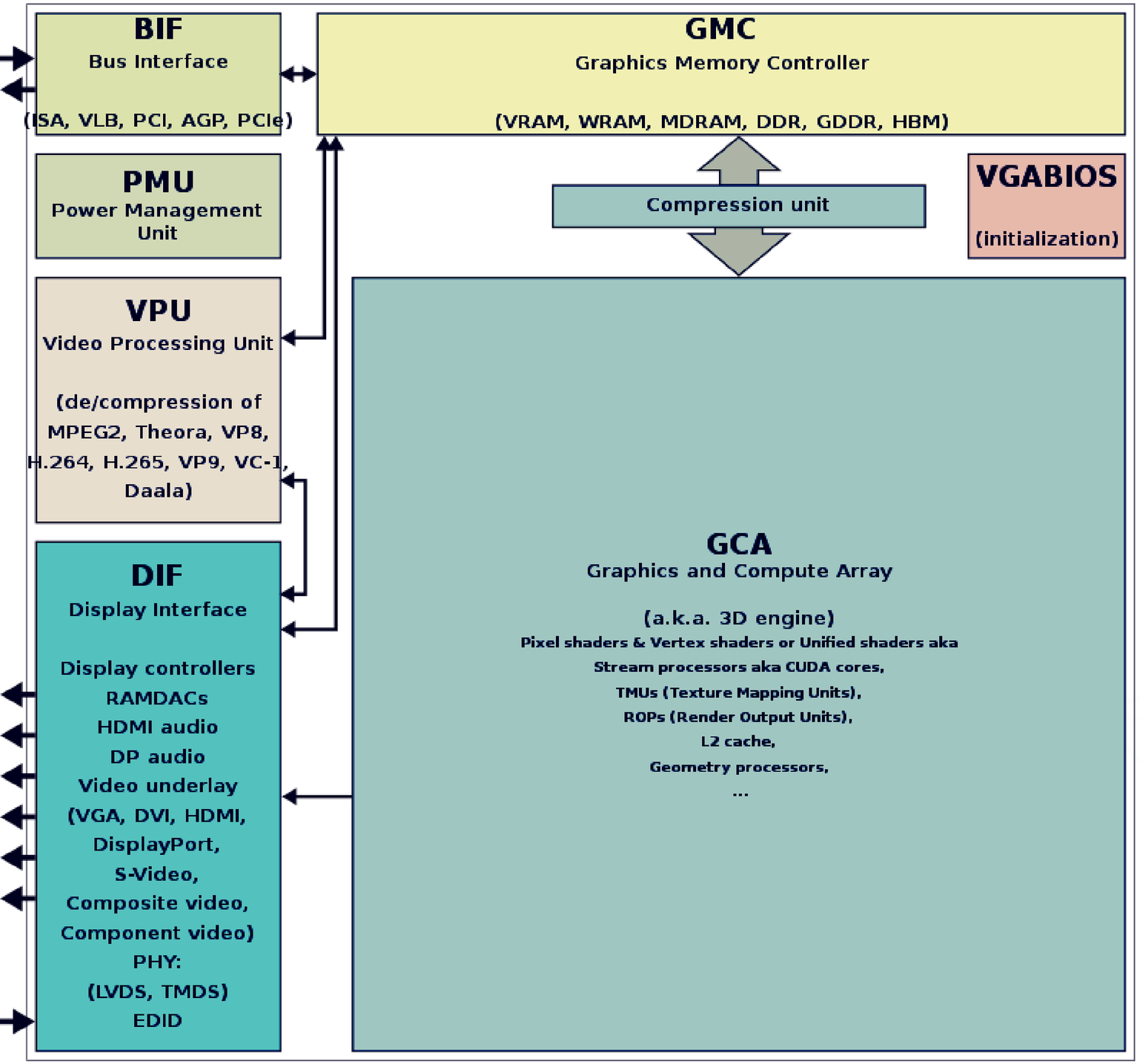


DIAGRAM OF GPU ARCHITECTURE

References

[1] https://nvidia.custhelp.com/app/answers/detail/a_id/2132/~/what-is-cuda%3F

[2] https://docs.nvidia.com/cuda/cuda-c-programming-guide/

[3] https://developer.nvidia.com/blog/easy-introduction-cuda-c-and-c/

[4] https://developer.nvidia.com/cuda-downloads

[5] https://www.partitionwizard.com/partitionmagic/gpu-architecture.html

[6] https://www.cdw.com/content/cdw/en/articles/hardware/cpu-vs-gpu.html

[7] https://core.vmware.com/resource/exploring-gpu-architecture