ΕΠΛ 372: Παράλληλη Επεξεργασία

Introduction to OpenCL Programming

Εργαστήριο 11

Πέτρος Παναγή, PhD

References:

https://www.khronos.org/opencl/

What is OpenCL?

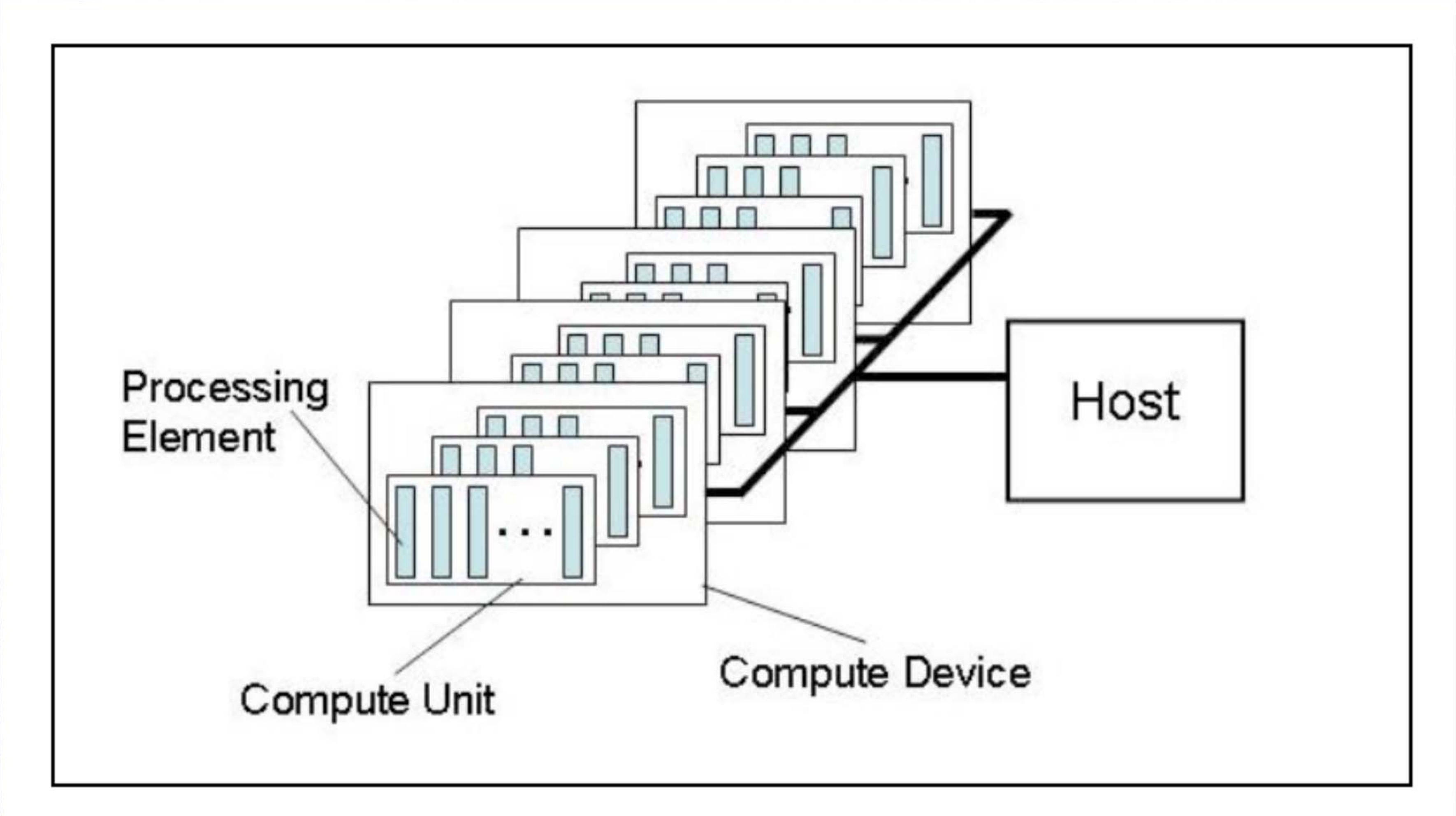
OpenCL is an **open industry standard** for programming a heterogeneous collection of **CPUs, GPUs** and other discrete computing devices organized into a single platform.

It is more than a language.

OpenCL is a framework for parallel programming and includes a language, API, libraries and a runtime system to support software development.

Using OpenCL, for example, a programmer can write general purpose programs that execute on GPUs without the need to map their algorithms onto a 3D graphics API such as OpenGL or DirectX (known as GPGPU).

Platform Model



Execution Model

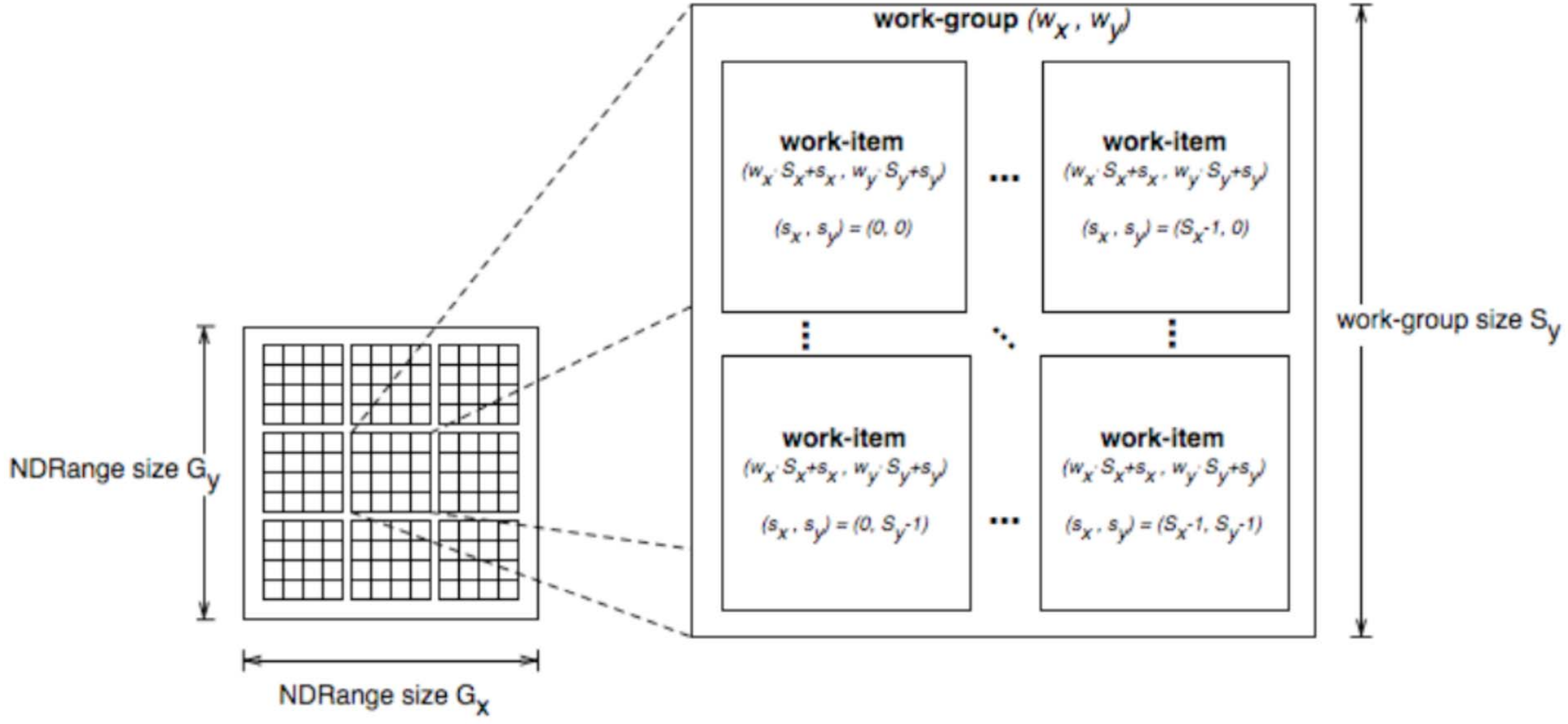
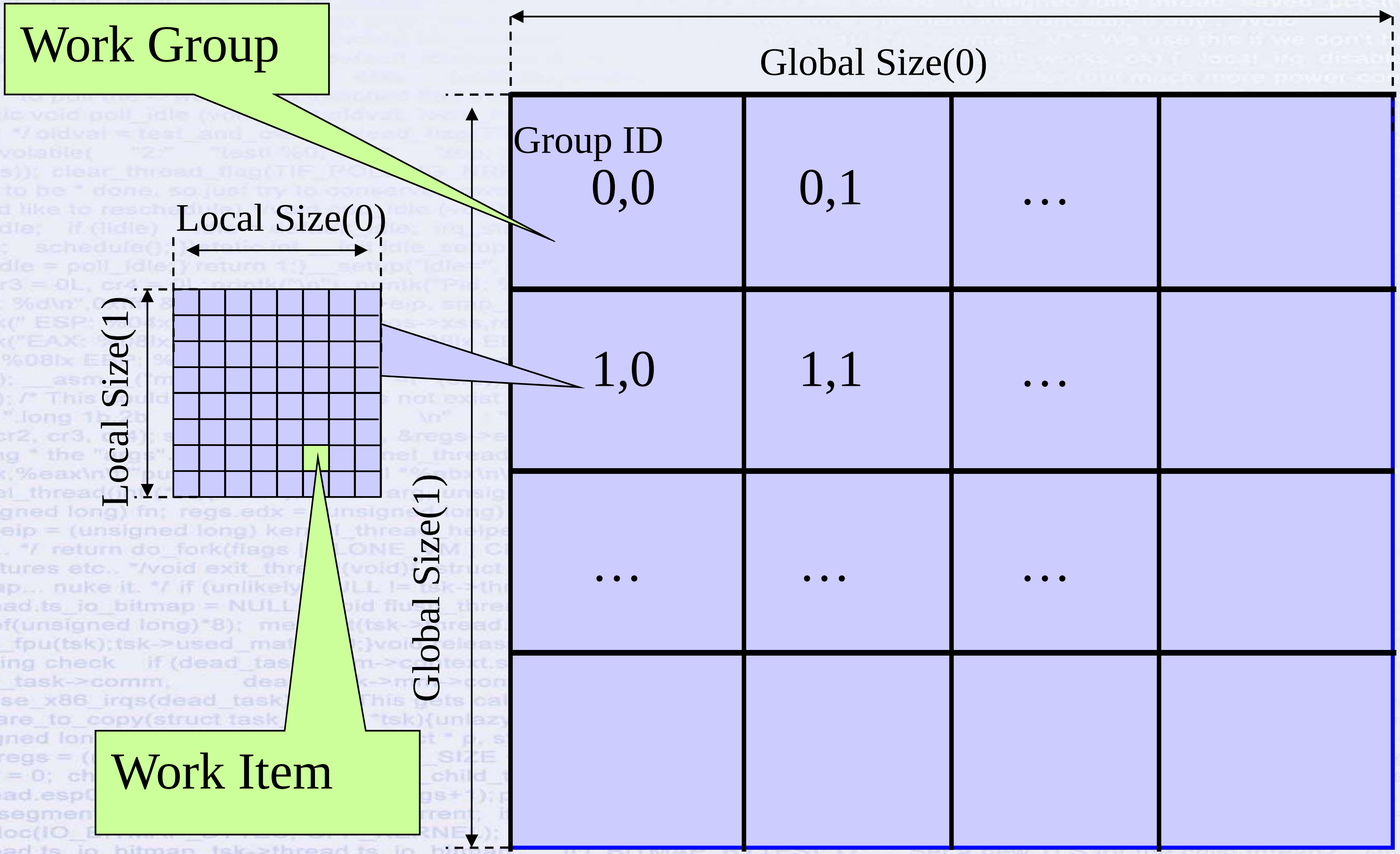

work-group (w_x, w_y)
work-item
$(w_x \cdot S_x+s_x, w_y \cdot S_y+s_y)$
$(s_x, s_y) = (0, 0)$
work-item
$(w_x \cdot S_x+s_x, w_y \cdot S_y+s_y)$
$(s_x, s_y) = (S_x\text{-}1, 0)$
work-item
$(w_x \cdot S_x+s_x, w_y \cdot S_y+s_y)$
$(s_x, s_y) = (0, S_y\text{-}1)$
work-item
$(w_x \cdot S_x+s_x, w_y \cdot S_y+s_y)$
$(s_x, s_y) = (S_x\text{-}1, S_y\text{-}1)$
work-group size S_y
NDRange size G_y
NDRange size G_x

Figure 3.2 *An example of an NDRange index space showing work-items, their global IDs and their mapping onto the pair of work-group and local IDs.*

OpenCL NDRange Configuration



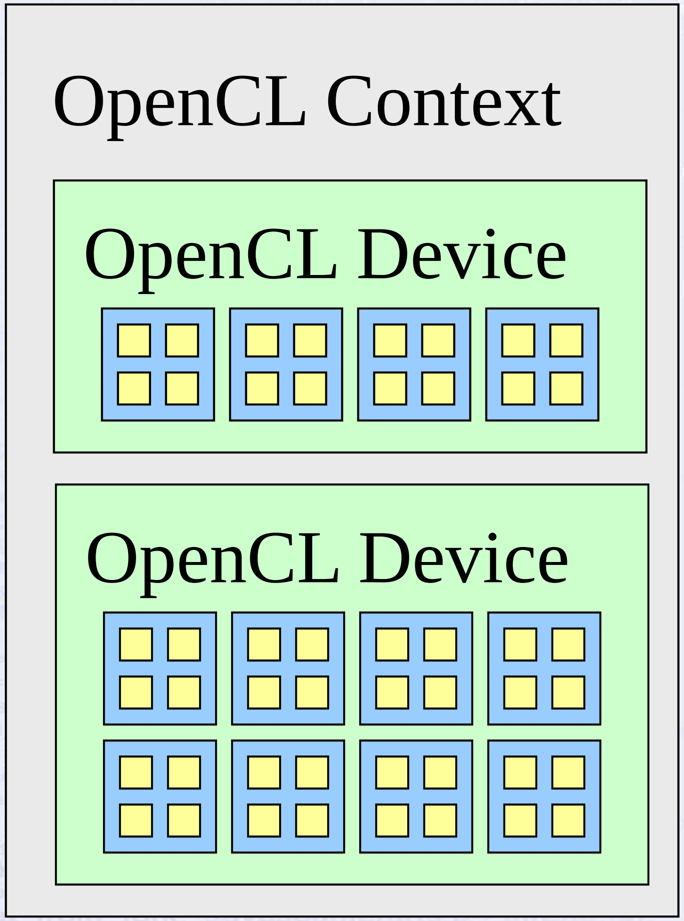
Execution Model: Context and Command Queues

The **host** defines a **context** for the execution of the **kernels**. The context includes the following resources:

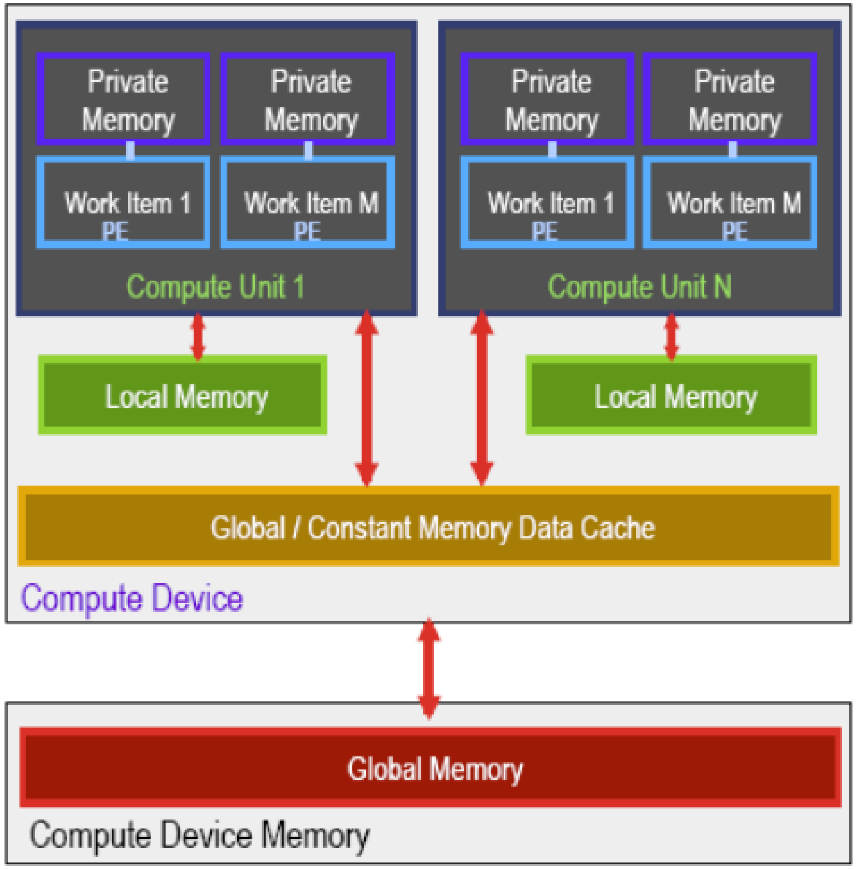
1. **Devices**: The collection of OpenCL devices to be used by the host. (The things that are doing the execution)
2. **Kernels:** The OpenCL functions that run on OpenCL devices.
3. **Program Objects:** The program source or executable that implement the kernels.
4. **Memory Objects:** A set of memory objects visible to the host and the OpenCL devices. Memory objects contain values that can be operated on by instances of a kernel.
5. **Command queues:** mechanisms for interaction with the devices

OpenCL Context

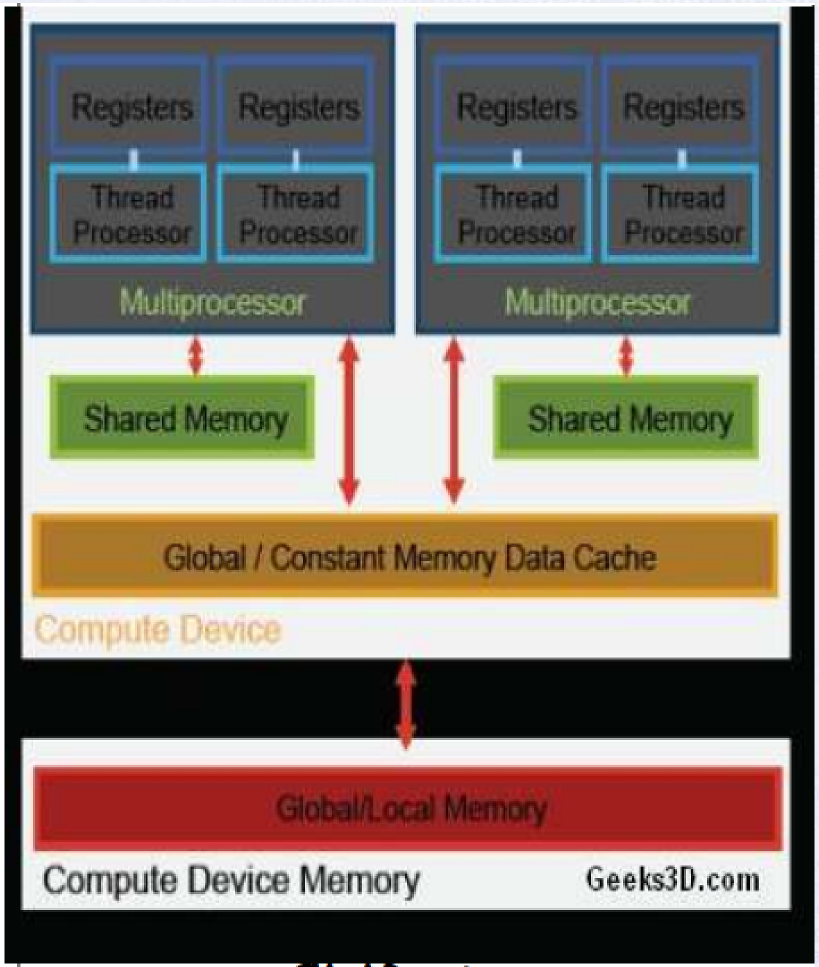
- Contains one or more devices
- OpenCL memory objects are associated with a context, not a specific device
- **clCreateBuffer()** emits error if an allocation is too large for any device in the context
- Each device needs its own **work queue**(s)
- Memory transfers are associated with a **command queue** (thus a specific device)



Memory Model Comparison



OpenCL



CUDA

OpenCL to CUDA Data Parallelism Model Mapping

OpenCL Parallelism Concept	CUDA Equivalent
kernel	kernel
host program	host program
NDRange (index space)	grid
work item	thread
work group	block

OpenCL device architecture

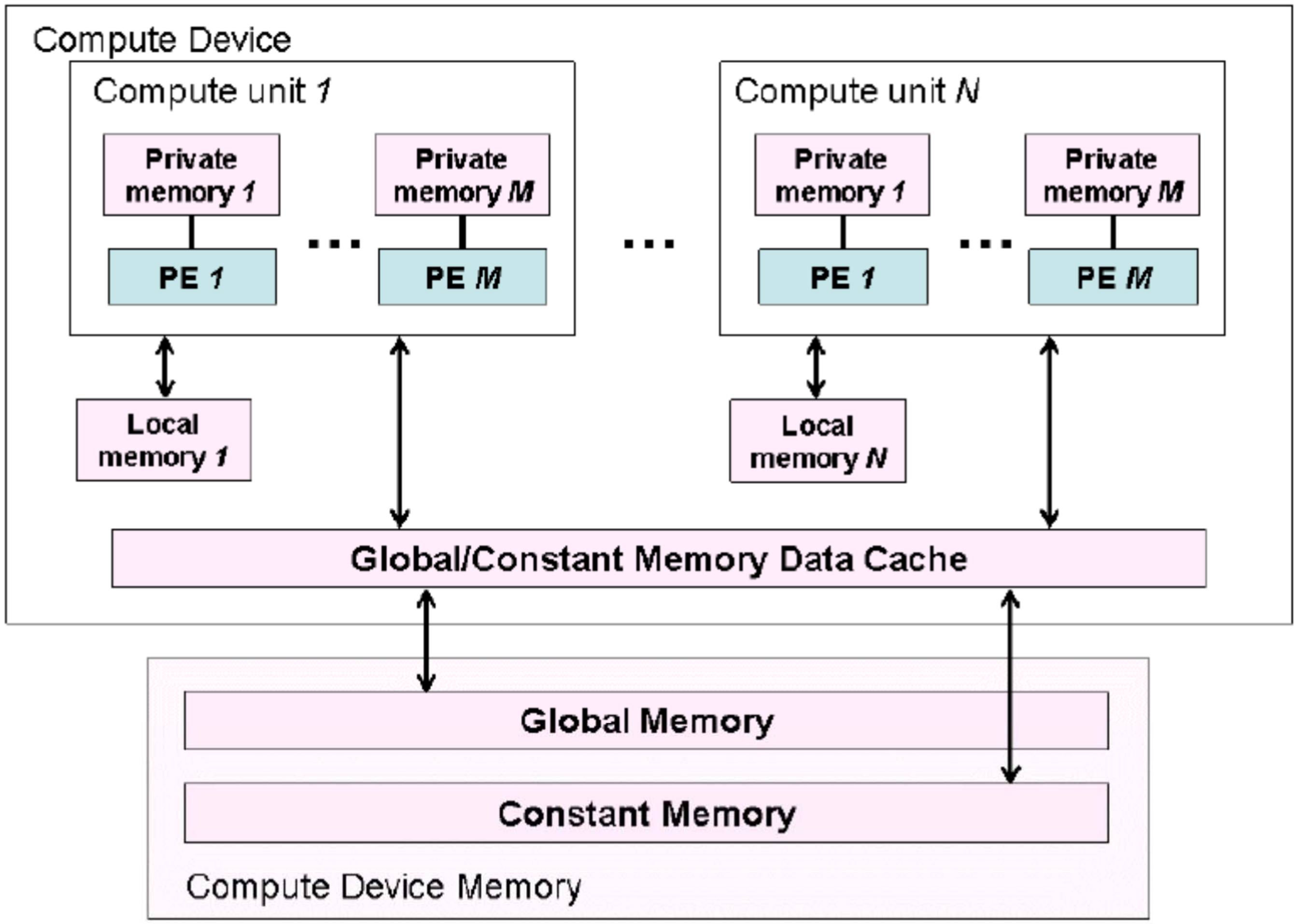


Figure 3.3: *Conceptual OpenCL device architecture with processing elements (PE), compute units and devices. The host is not shown.*

Mapping OpenCL Memory Types to CUDA

OpenCL Memory Types	CUDA Equivalent
global memory	global memory
constant memory	constant memory
local memory	shared memory
private memory	local memory

Mapping of OpenCL Dimensions and Indices to CUDA

OpenCL API Call	Explanation	**CUDA Equivalent**
get_global_id(0);	global index of the work item in the x dimension	blockIdx.x×blockDim.x+threadIdx.x
get_local_id(0)	local index of the work item within the work group in the x dimension	blockIdx.x
get_global_size(0);	size of NDRange in the x dimension	gridDim.x ×blockDim.x
get_local_size(0);	Size of each work group in the x dimension	blockDim.x

Matrix Multiplication Code in OpenCL

Check the code from the web Page.