VRIJE UNIVERSITEIT BRUSSEL

ULB UNIVERSITÉ LIBRE DE BRUXELLES

GPU Computing

» Lesson 7: Warps & SIMT

Gauthier Lafruit & Jan Lemeire

2022-2023

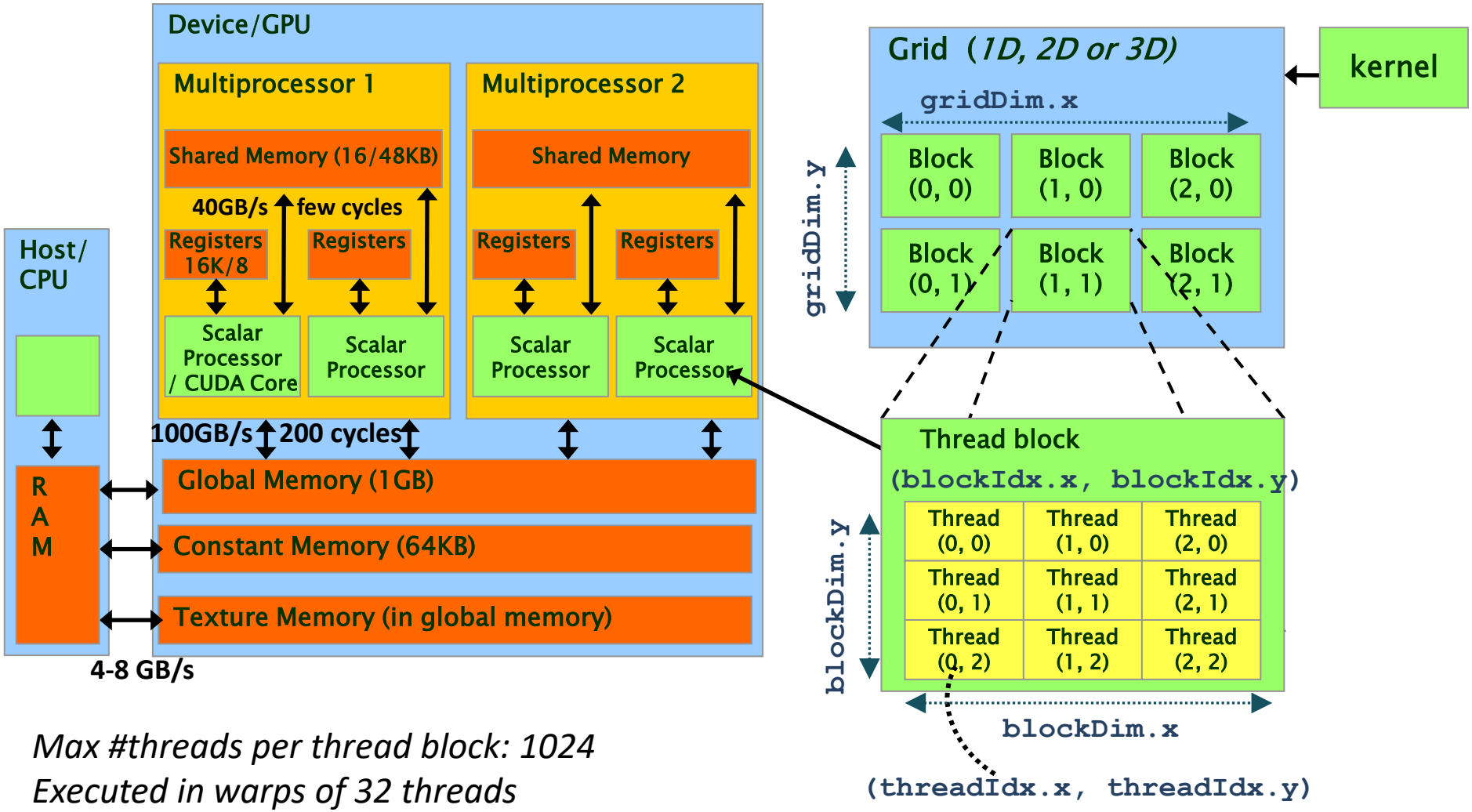
http://parallel.vub.ac.be/education/gpu

VRIJE UNIVERSITEIT BRUSSEL

Levels of Understanding

- Level 0
  - Host code
- Level 1
  - Parallel execution on the device
- Level 2
  - Device model and work groups
- Level 3 *=> explained here*
  - Hardware threads & SIMT

GPU Concepts



Max #threads per thread block: 1024
Executed in warps of 32 threads
Max thread blocks simultaneously on MP: 8
Max active warps on MP: 24/48

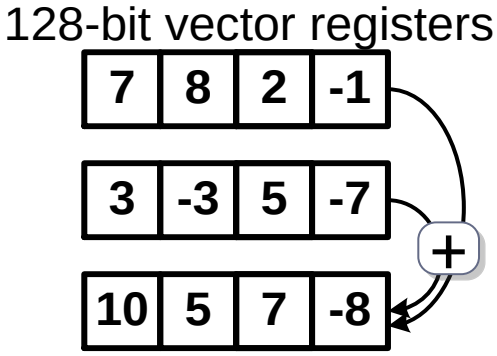
CUDA terminology

Vector processors & SIMD

One way to do several computations at the same time

VRIJE UNIVERSITEIT BRUSSEL

Vector processors



Instructions are performed at once on all elements of the vector registers

- All processing elements execute the same instruction at the same time
  - Multiple data elements in 128-bit or 256-bit wide registers (vector registers)
  - MMX and SSE instructions in x86
- Instead of iterating over the vector (for-loop), one instruction is sufficient

VRIJE UNIVERSITEIT BRUSSEL

Vector processors

- Simplifies synchronization
- Reduced instruction control hardware: an instruction has to be read only once for x number of calculations
- Works best for highly data-parallel applications
- Has long be viewed as the solution for high-performance computing
  - Why always repeating the same instructions (on different data)? => just apply the instruction immediately on all data
- *However*: difficult to program, since less flexible
  - Is OpenCL/SIMT easier?

VRIJE UNIVERSITEIT BRUSSEL

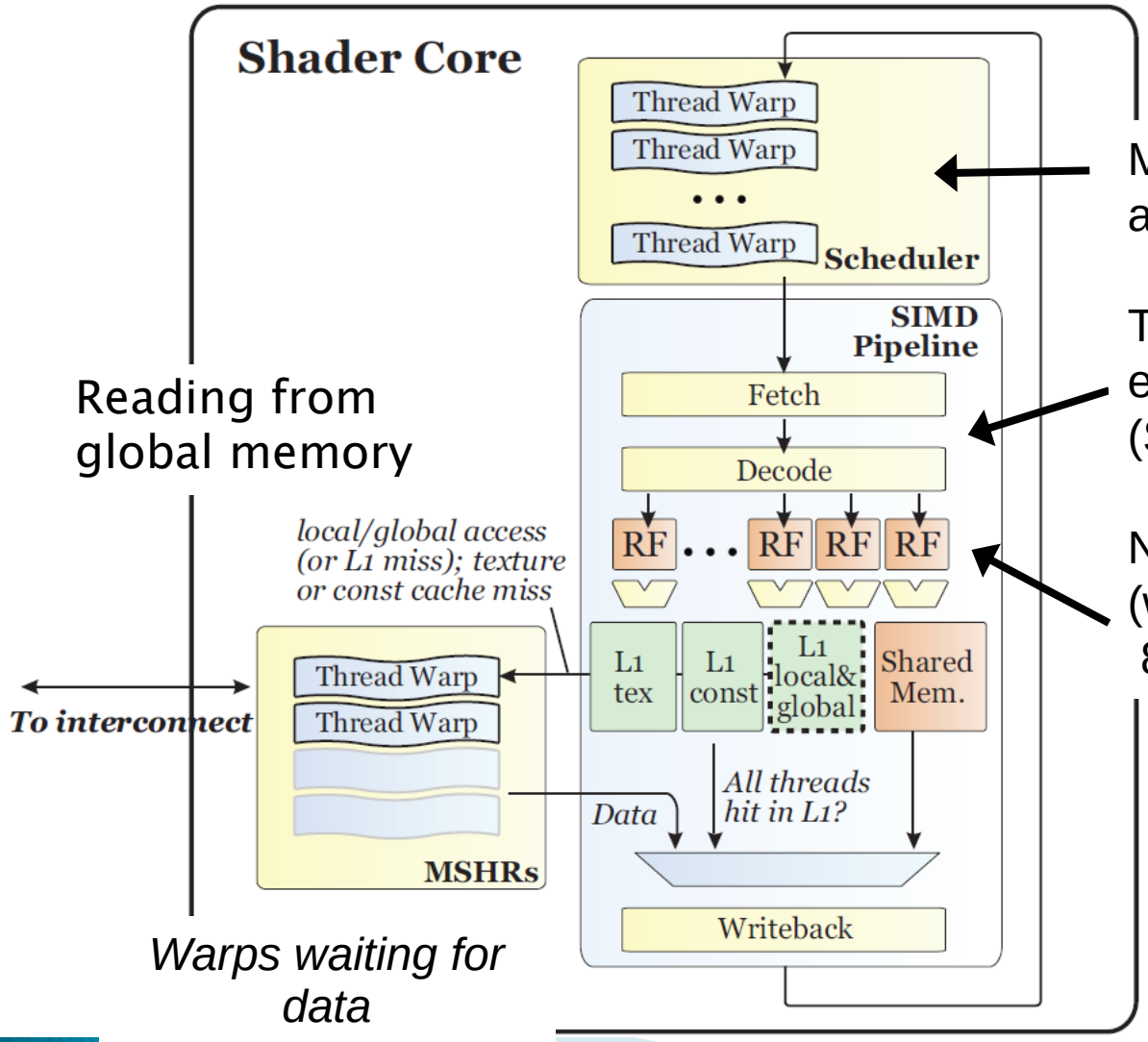
Instruction and Data Streams

		Data Streams	
		Single	Multiple
Instruction Streams	Single	**SISD**: Intel Pentium 4	**SIMD**: Vector instructions of x86
	Multiple	**MISD**: No examples today	**MIMD**: Intel Xeon e5345

- Vector processing = Single Instruction on Multiple Data (SIMD)

Level 3
Hardware Threads & SIMT

1 Streaming Multiprocessor = a pipeline



Multiple warps (hardware threads) are simultaneously active

The Same Instruction is executed on Multiple Threads (SIMT)

Number of processing elements (width of pipeline):
8 – 32 – 192 - 128

VRIJE UNIVERSITEIT BRUSSEL

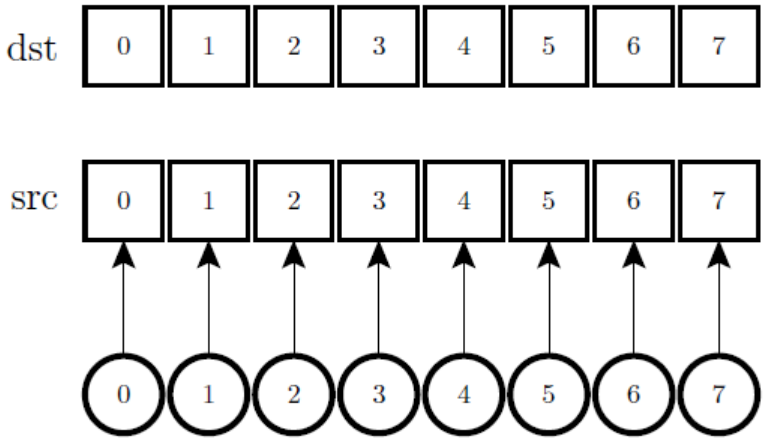
Warp executes kernel threads in lock step

- **Hardware thread (called warp by Nvidia):**
  - Kernel threads are executed together in groups, the instructions of the kernel are executed at the same time they will execute the same instruction
  - Nvidia: 32; AMD: 64; Intel: variable number (8/16/24/32)
- **Consequences:**
  1. Running 1 kernel thread or 32 kernel threads takes the same amount of time
     - Thus: create thread blocks which are multiples of 32 or 64
  2. **Branching**: if kernel threads of the same warp take different branches, all branches will be executed after each other
     - Performance loss
  3. **Concurrent memory access**: if kernel threads access memory, all kernel threads of the same warp do it simultaneously
     - Not all memory access can be done with the same speed

VRIJE UNIVERSITEIT BRUSSEL

When is SIMT = vector processing?

- Contiguous data access (See lesson 2)



- In this case, warp execution of instructions on the data is similar to vector instructions operating on vector registers.

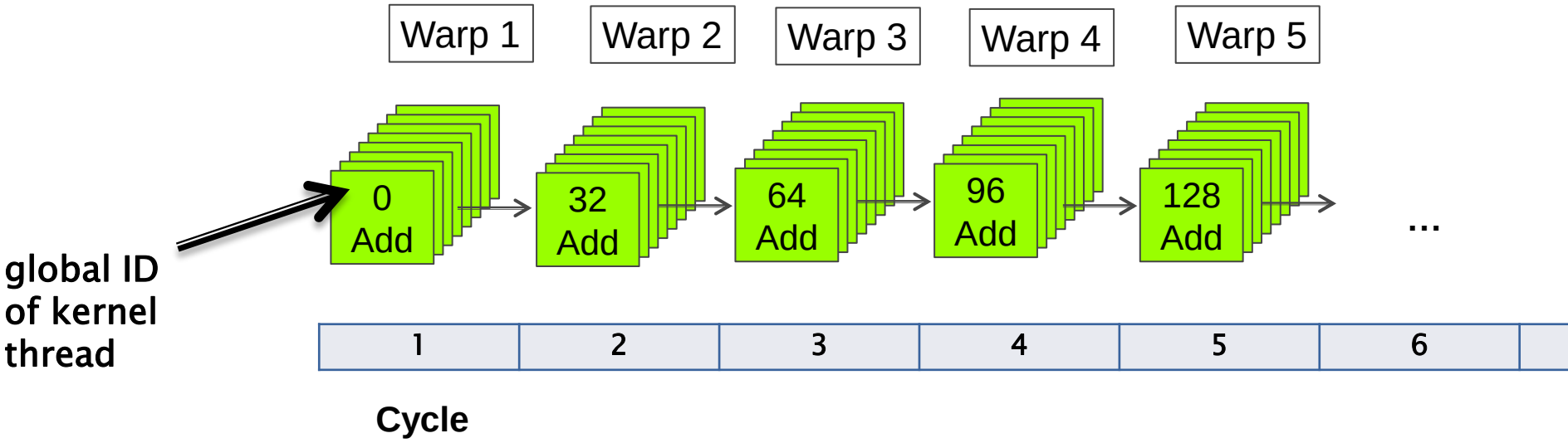
VRIJE UNIVERSITEIT BRUSSEL

Vectors versus SIMT

- **Vectors (SIMD)**
  - Data should be stored in vector register
  - Instructions are performed onto these registers
  - Harder to program
- **SIMT**
  - Each thread of a warp can choose on which data it works
  - Easier to program: programmer does not have to worry about *thread-data mapping*

Warp execution

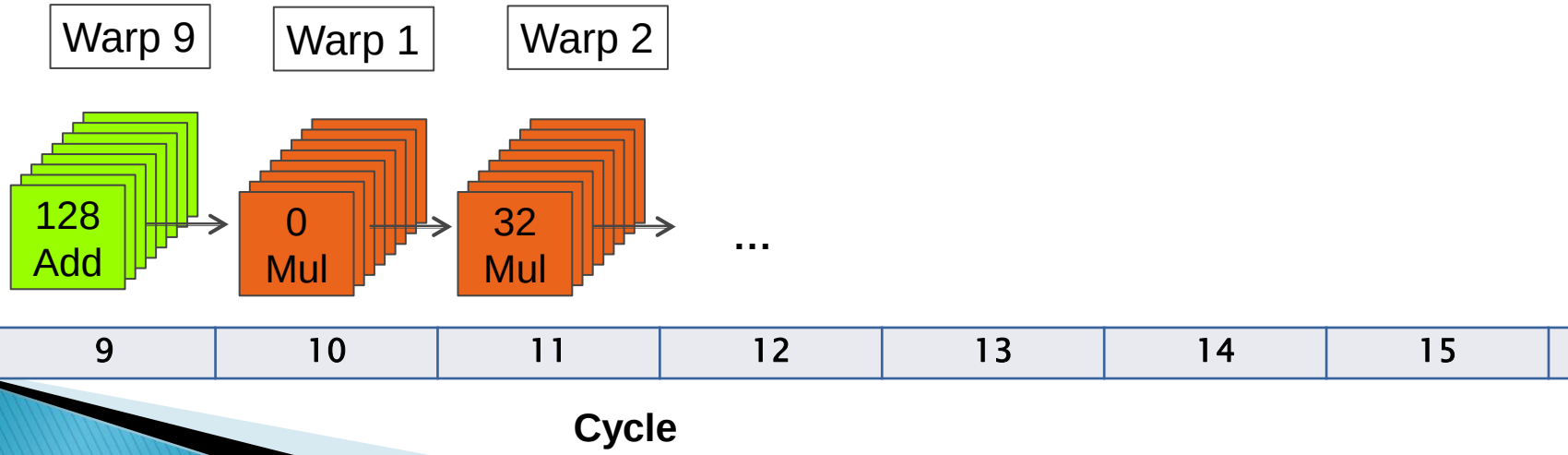
- Kernel threads are sent into the pipeline grouped in a warp
  - ALUs all execute the same instruction in `lockstep': Single Instruction, Multiple Threads (SIMT)
  - Every cycle a new warp can issue an instruction



Warp execution

- On an Nvidia Kepler architecture, a single precision floating point instruction (add or multiplication) takes 9 cycles, which is the depth of the pipeline.
  - 8 other warps can be scheduled in the mean time
  - After 9 cycles, the second instruction of the first warp (multiplication) can be issued, next the second warp and so on

  ⇒With 9 warps the pipeline is completely filled, no stalling/idling, the completion latency of 9 cycles is completely hidden.



VRIJE UNIVERSITEIT BRUSSEL

SIMT Conditional Processing

- If kernel threads of a warp follow different branches, the instructions of both branches have to be executed, but are desactivated for some threads.

  => Performance loss!

- **Example**: assume 8 threads, one instruction in if-clause, one in then-clause
  - 3 cyles in which 24 instructions are executed, 8 lost cycles (66% usage)