Introducción a CUDA

Miller Mendoza [1,2] y José Daniel Muñoz [1]

[1] Grupo Simulación de Sistemas Físicos, CEIBA-Complejidad
Departmento de Física, Universidad Nacional de Colombia,
Crr 30 # 45-03, Ed. 404, Of. 348, Bogotá D.C., Colombia

[2] ETH-Zürich, Computational Physics for Engineering Materials,
Institute for Building Materials,
Schafmattstrasse 6, HIF, CH-8093 Zürich, Switzerland

COLCIENCIAS

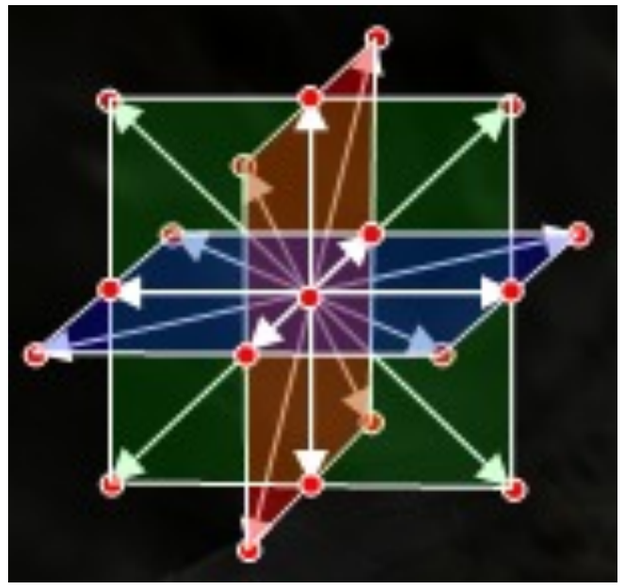
UNIVERSIDAD NACIONAL DE COLOMBIA

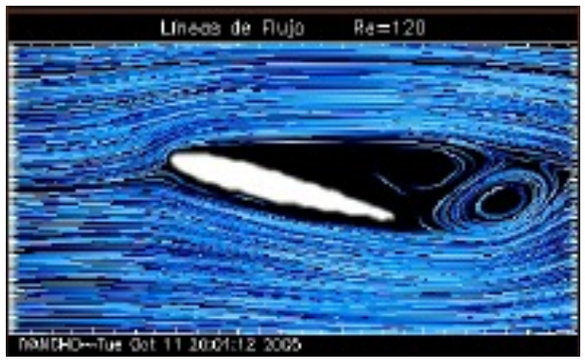
Introducción a CUDA

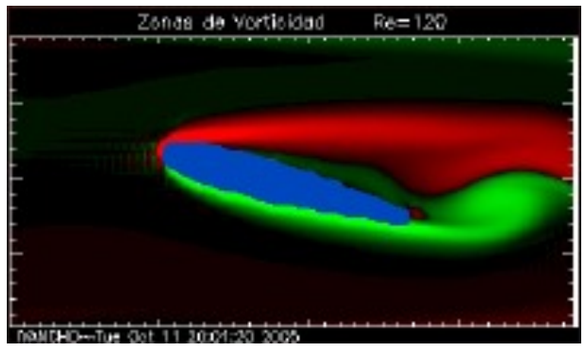
Miller Mendoza [1,2] y José Daniel Muñoz [1]

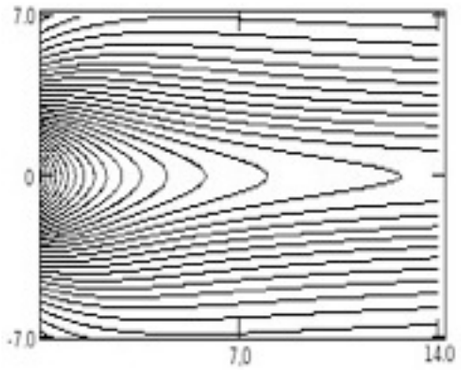
[1] Grupo Simulación de Sistemas Físicos, CEIBA-Complejidad
Departmento de Física, Universidad Nacional de Colombia,
Crr 30 # 45-03, Ed. 404, Of. 348, Bogotá D.C., Colombia

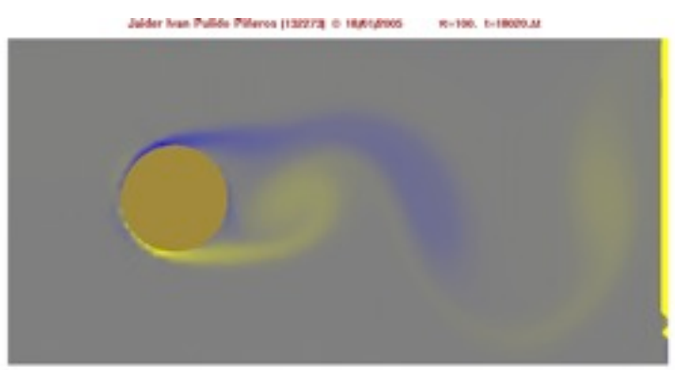
[2] ETH-Zürich, Computational Physics for Engineering Materials,
Institute for Building Materials,
Schafmattstrasse 6, HIF, CH-8093 Zürich, Switzerland







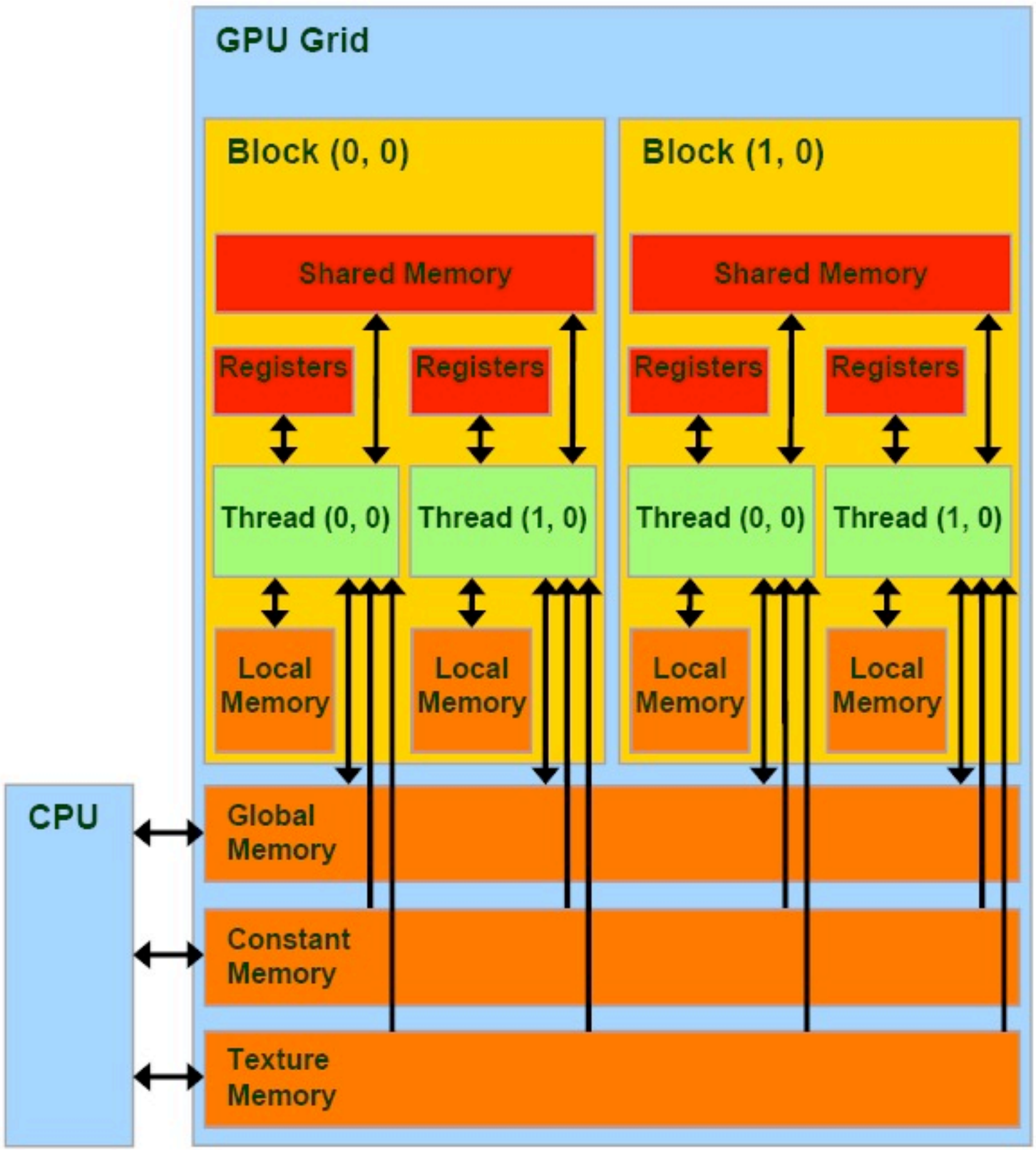




COLCIENCIAS

UNIVERSIDAD NACIONAL DE COLOMBIA

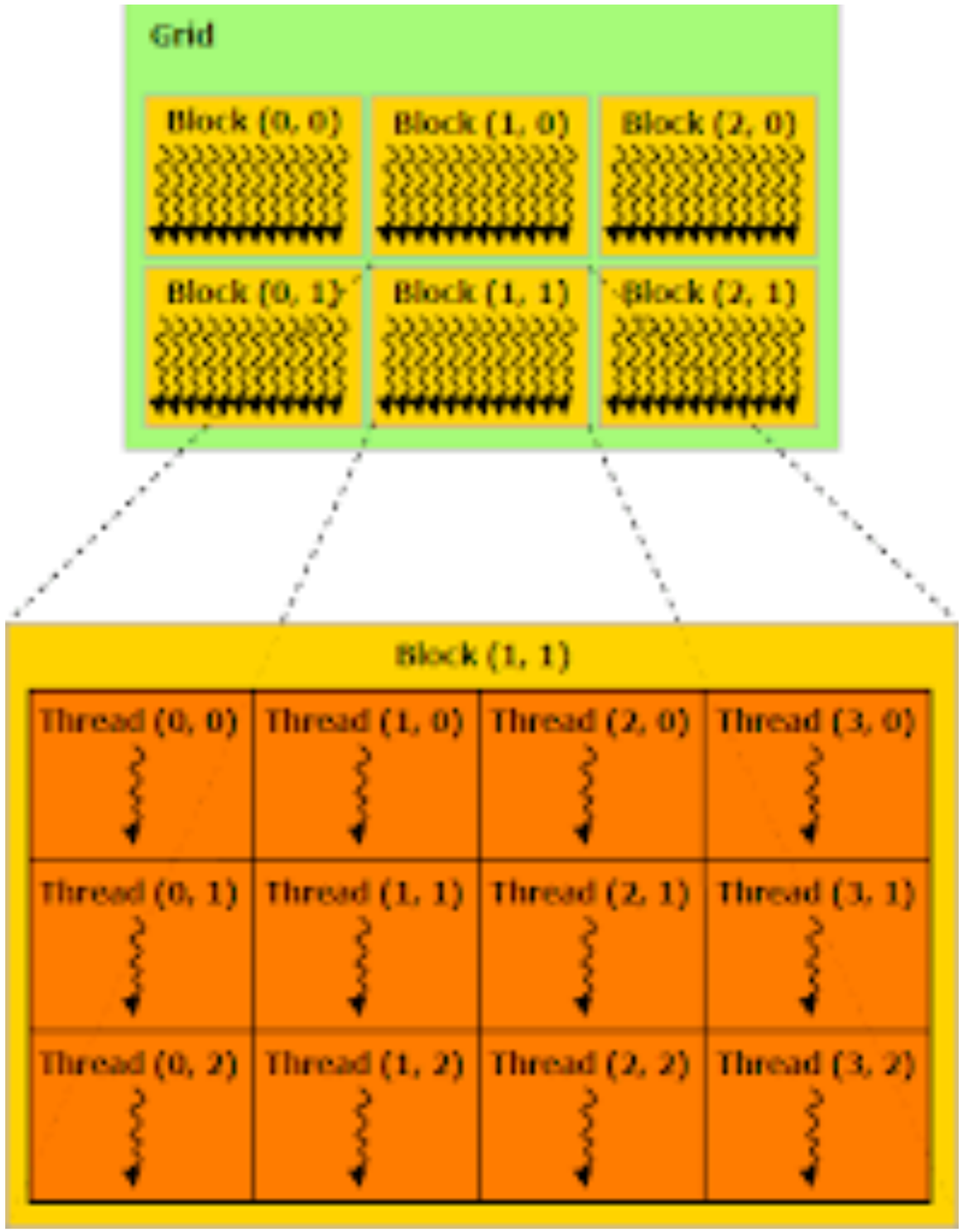
La tarjeta gráfica



Diferentes tipos de memoria

Entre más lejos, más grande, pero más lenta

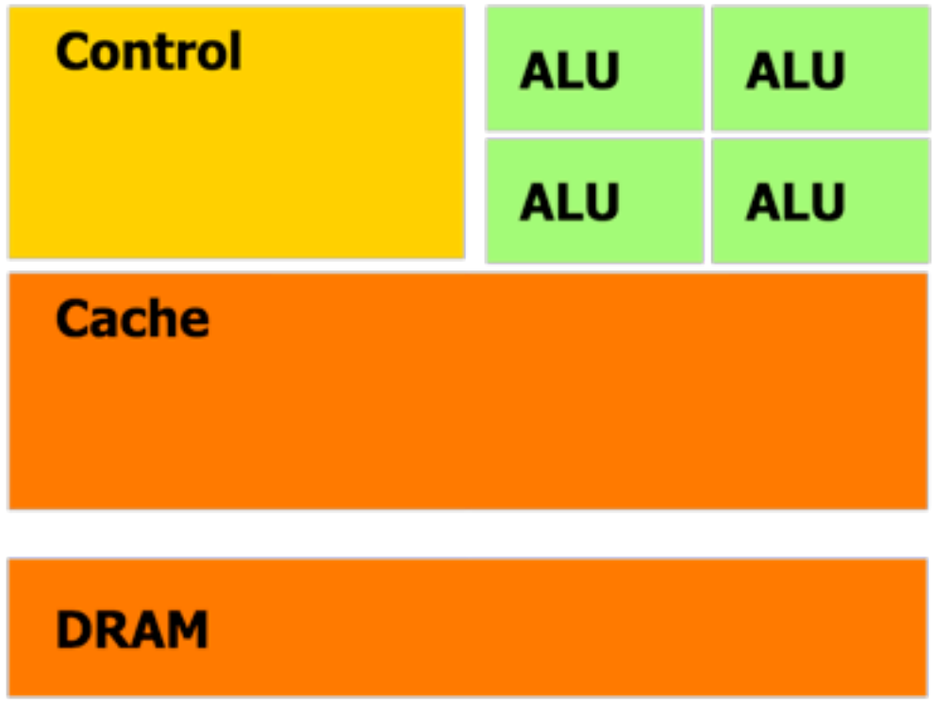
Grid, Blocks and Threads(hilos)

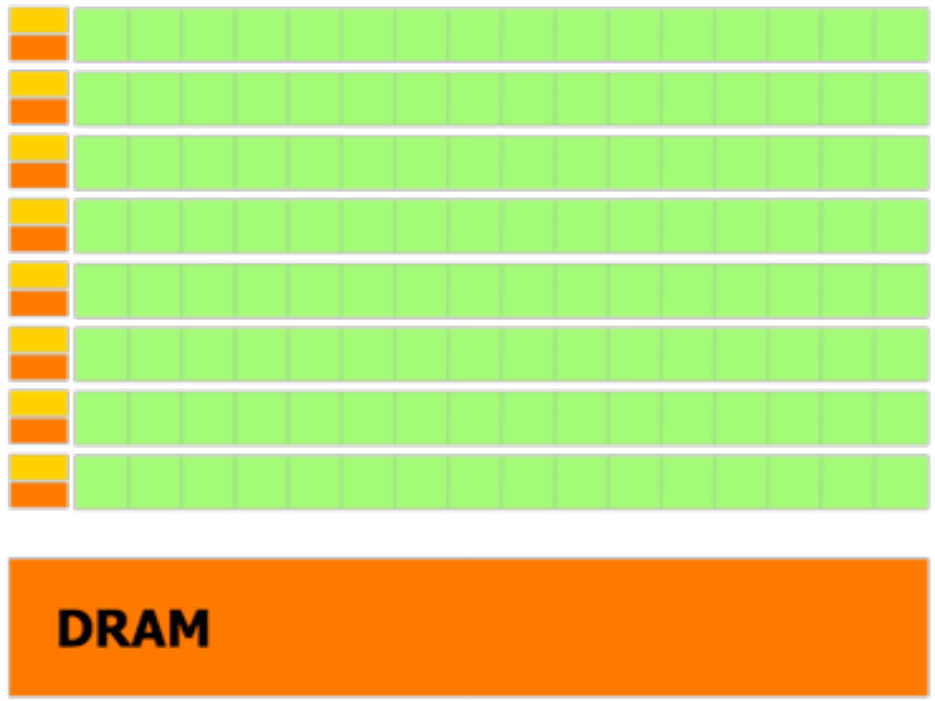


Las tareas se mandan en bloques de hilos

Cada hilo es lo que hace un procesador

Intercambio





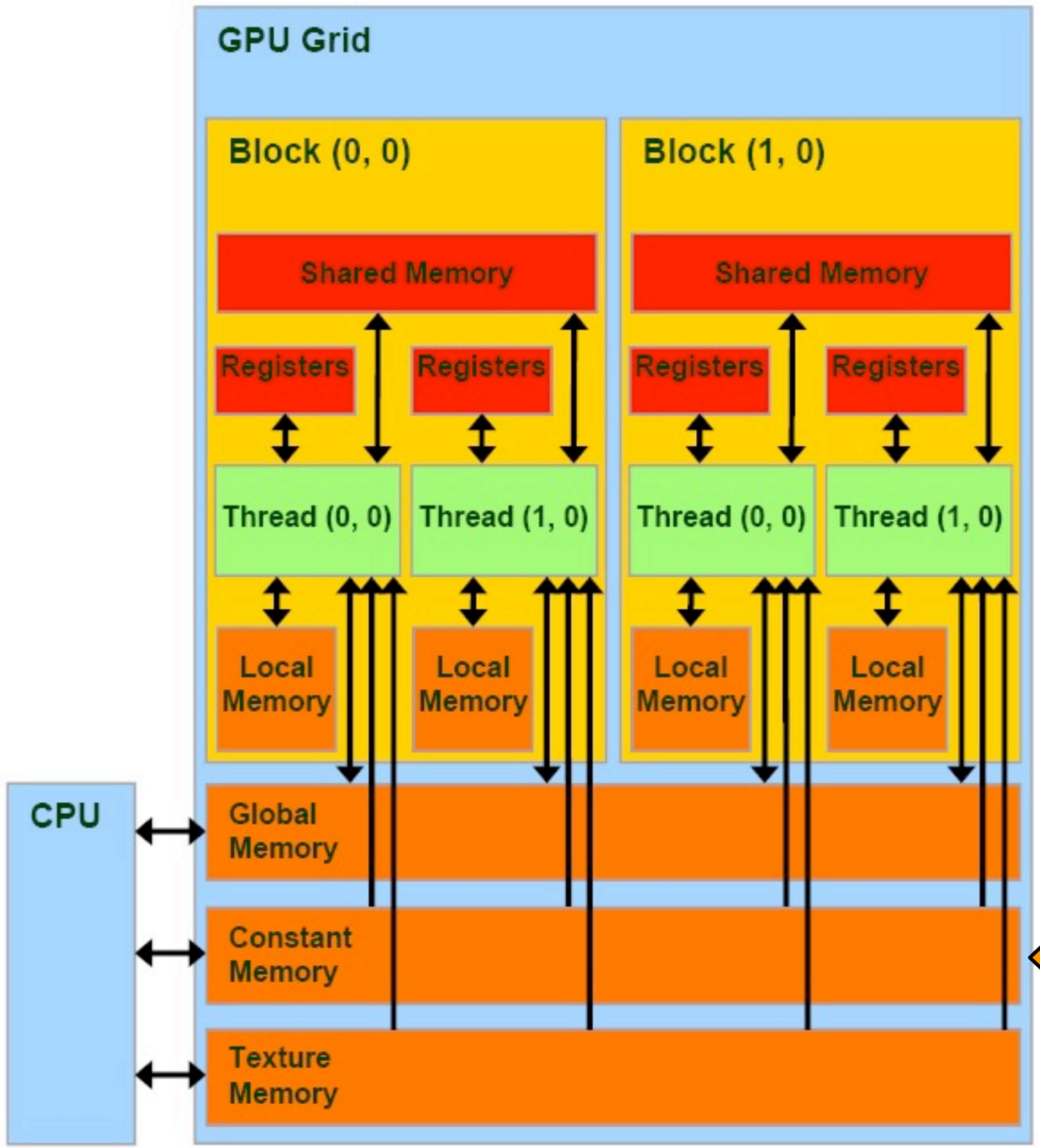
Host **Device**

Se mantiene una copia de los datos en cada uno, y se transfiere

Kernel=algo para que todos hagan.

- Mi primer programa en CUDA
- Suma de dos vectores en CUDA

Constantes en CUDA



Diferentes tipos de memoria

Pueden ser accesadas por todos los procesadores

Funciones en CUDA

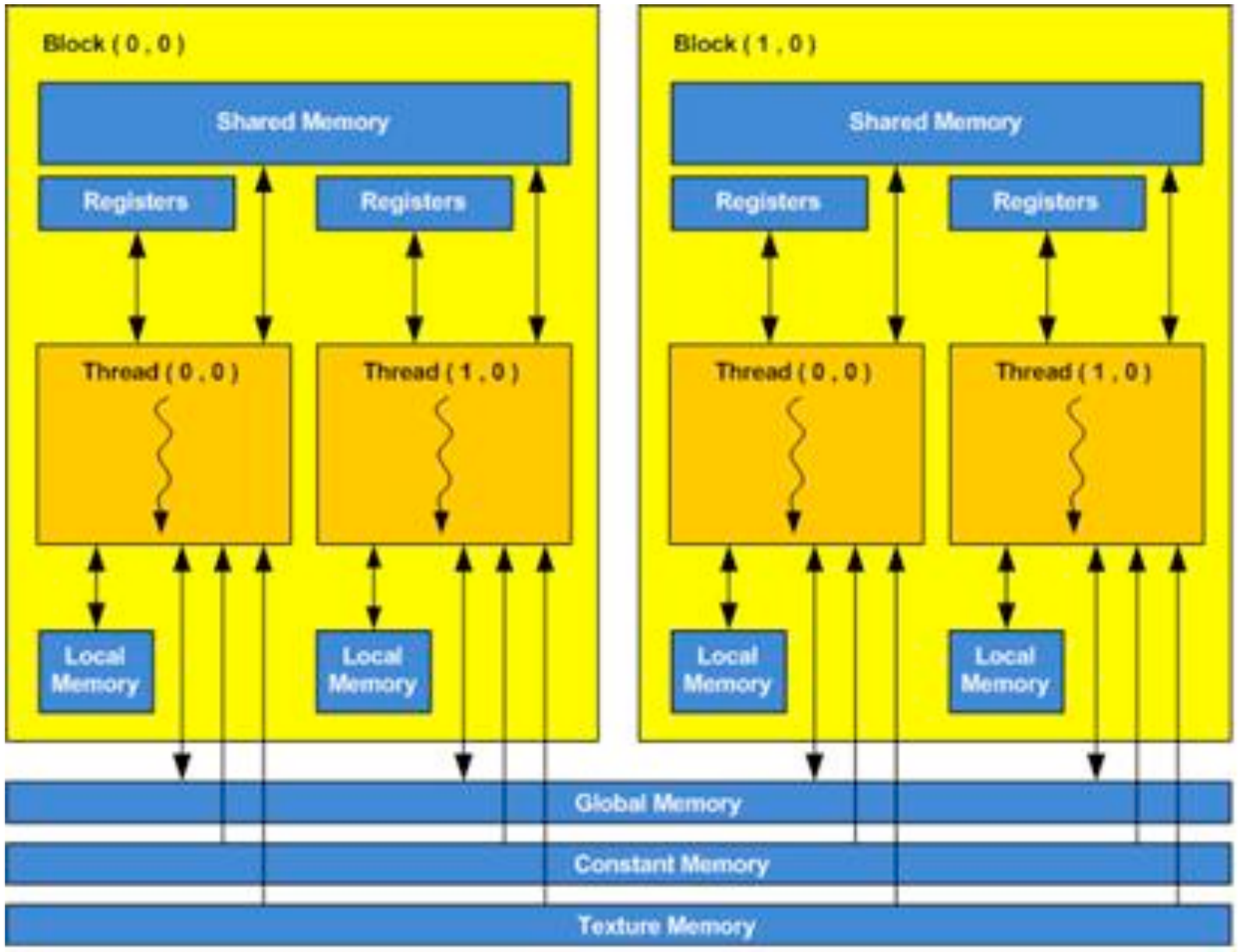
CUDA Function Declarations

	Executed on the:	Only callable from the:
`__device__ float DeviceFunc()`	device	device
`__global__ void KernelFunc()`	device	host
`__host__ float HostFunc()`	host	host

__host__ Es una orden a todos los procesadores
Debe ser void

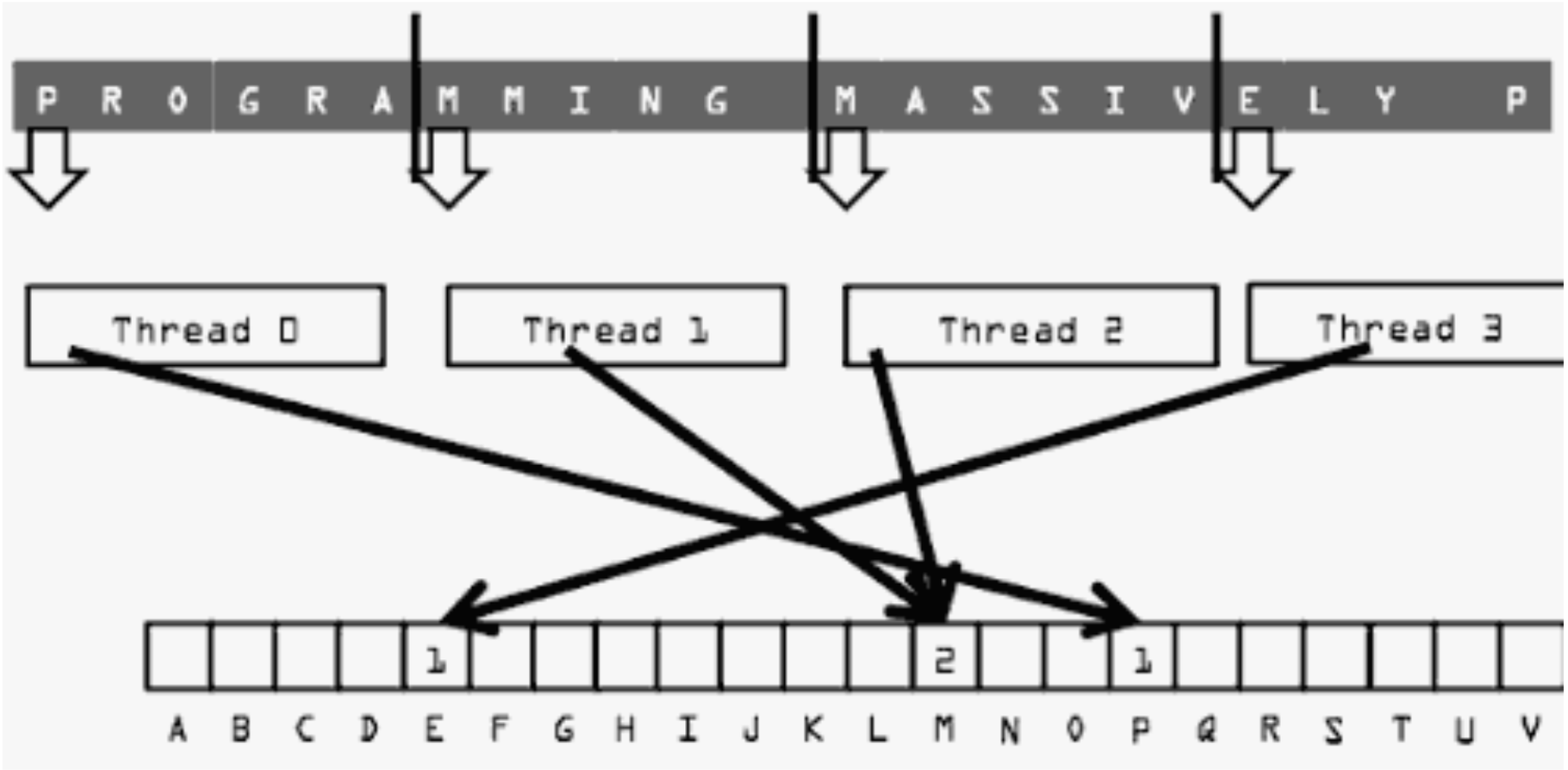
__device__ Es un cálculo parcial que hace cada procesador

Shared Memory en CUDA

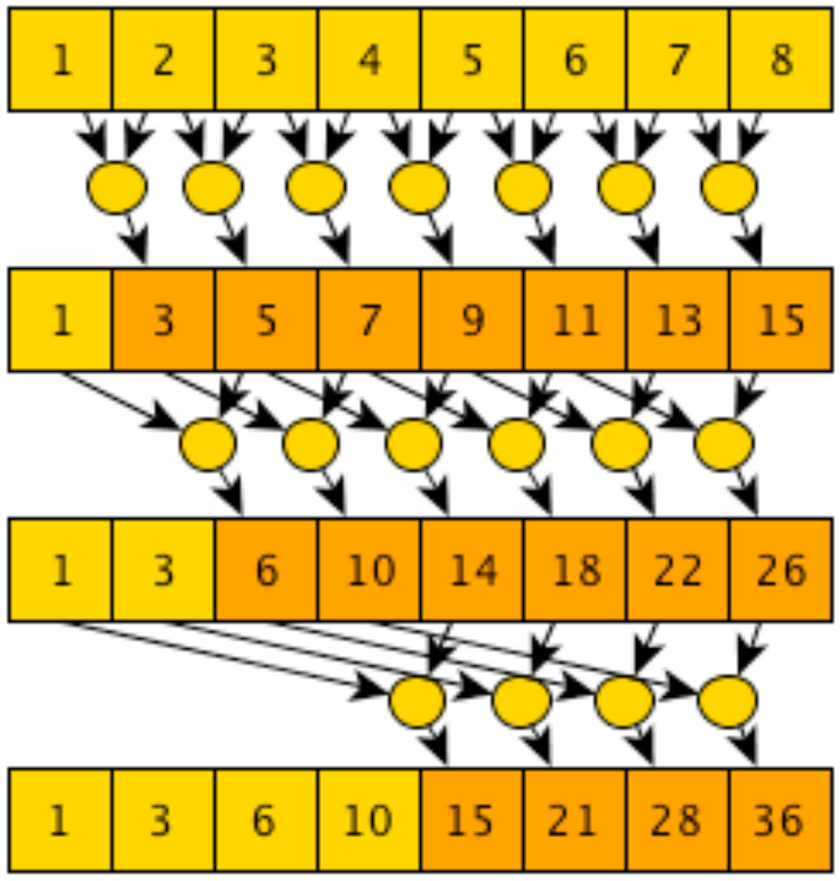


Es una memoria caché que todos los hilos de un mismo bloque pueden accesar

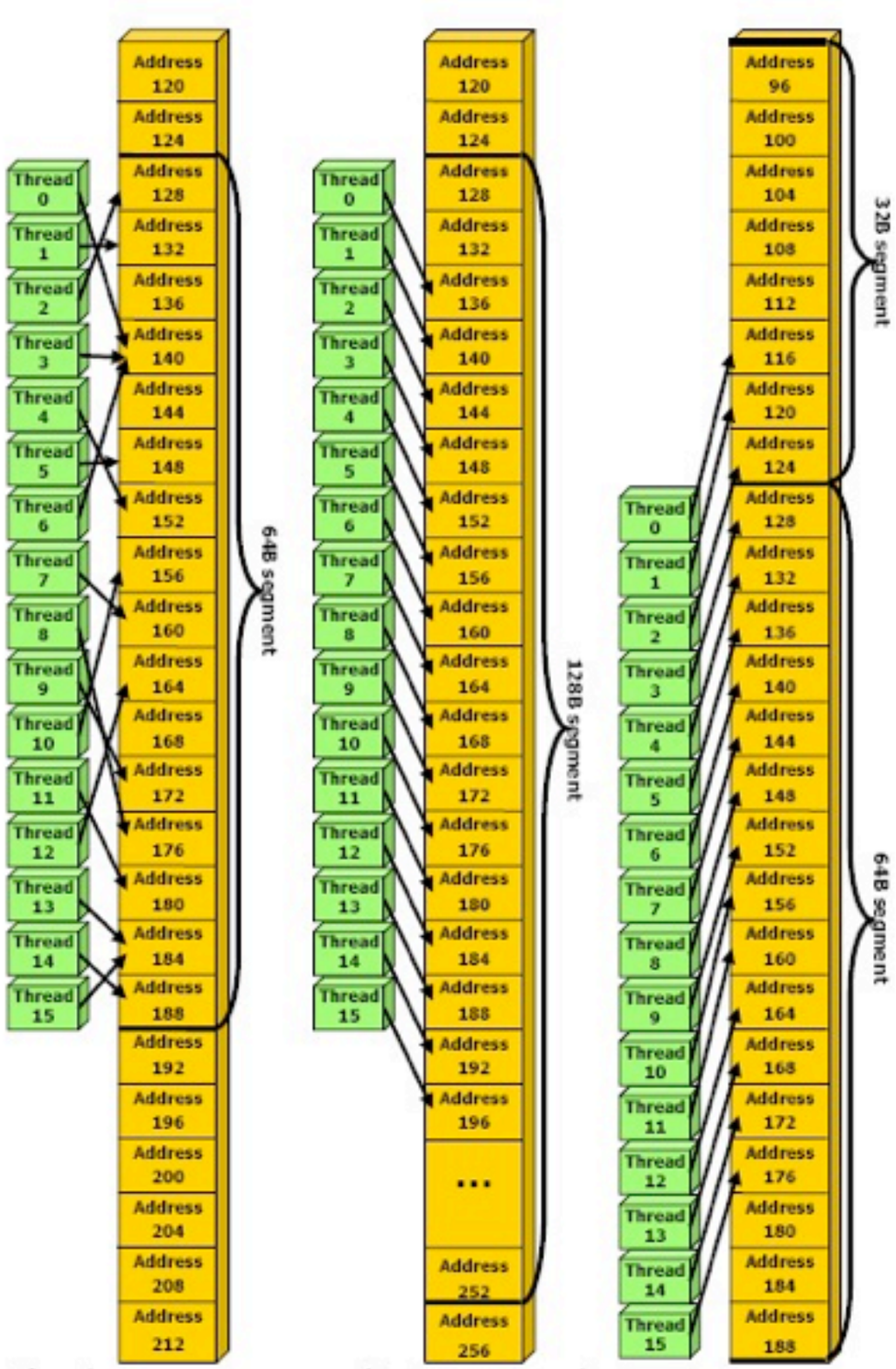
Atomic Operations in CUDA



Algoritmos en Paralelo



Matrices 2D y 3D: Pitch

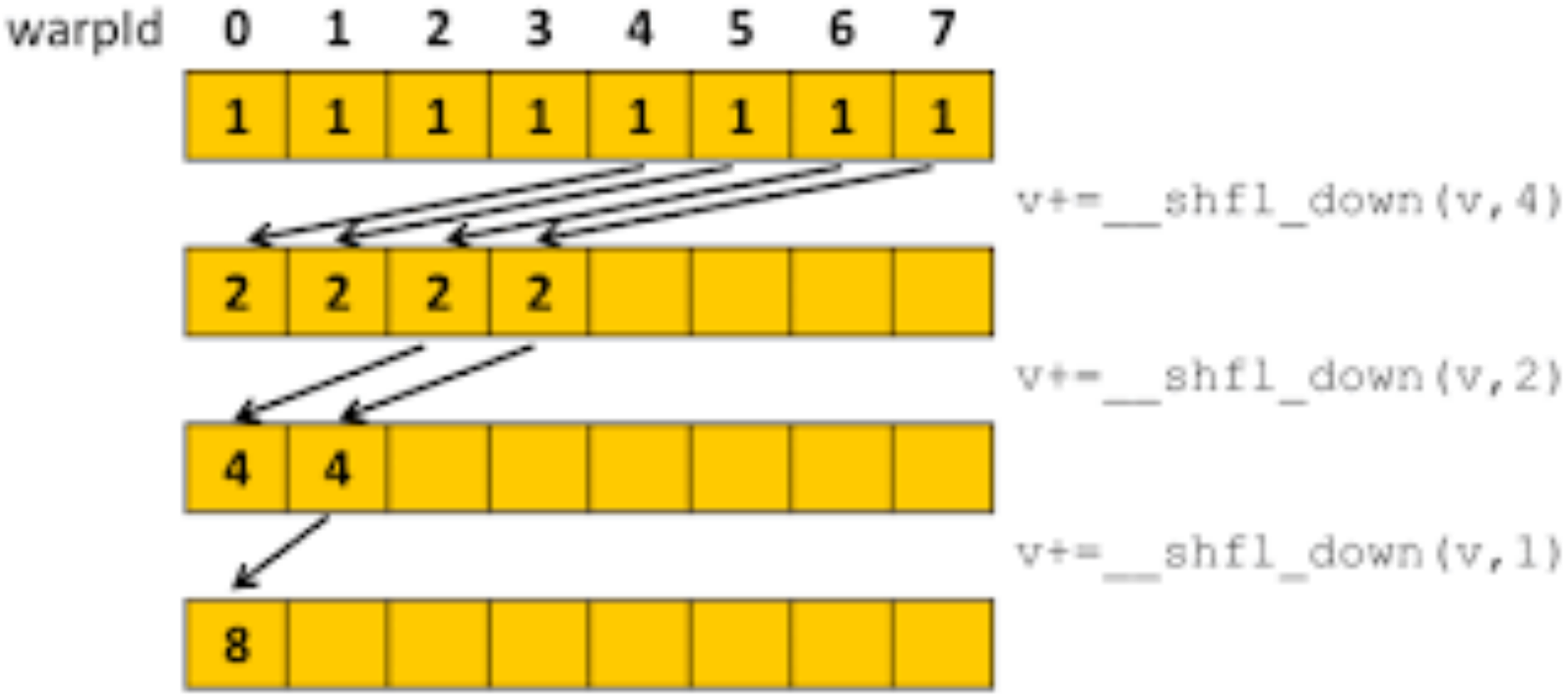


Left: random `float` memory access within a 64B segment, resulting in one memory transaction.
Center: misaligned `float` memory access, resulting in one transaction.
Right: misaligned `float` memory access, resulting in two transactions.

La memoria se transfiere en bloques enteros, y eso corre el inicio de la siguiente columna de la matriz

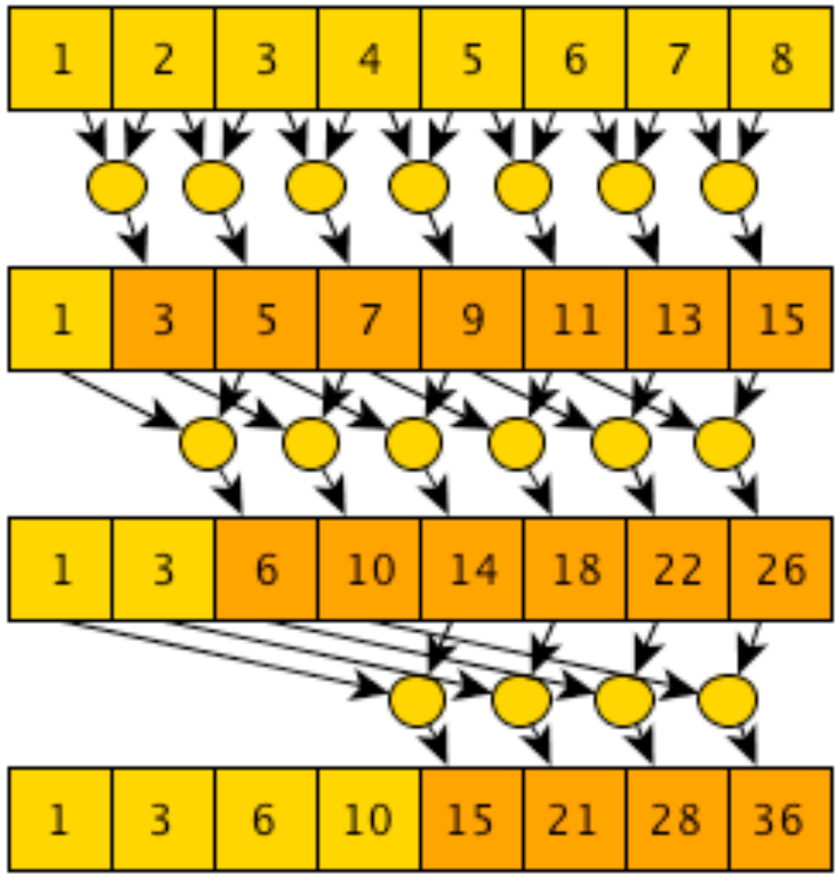
- Matriz 2D en CUDA

Reduce in CUDA

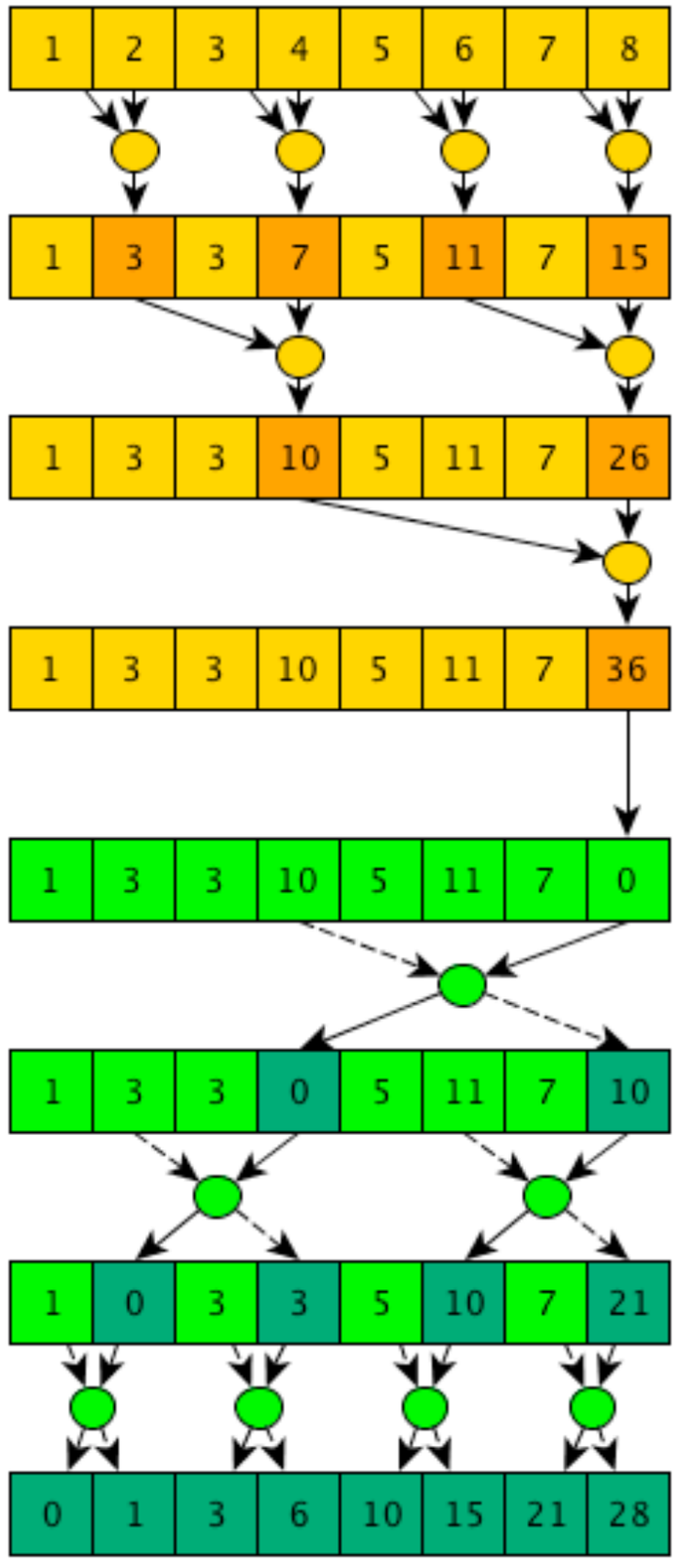


Scan

Hillis/Steele

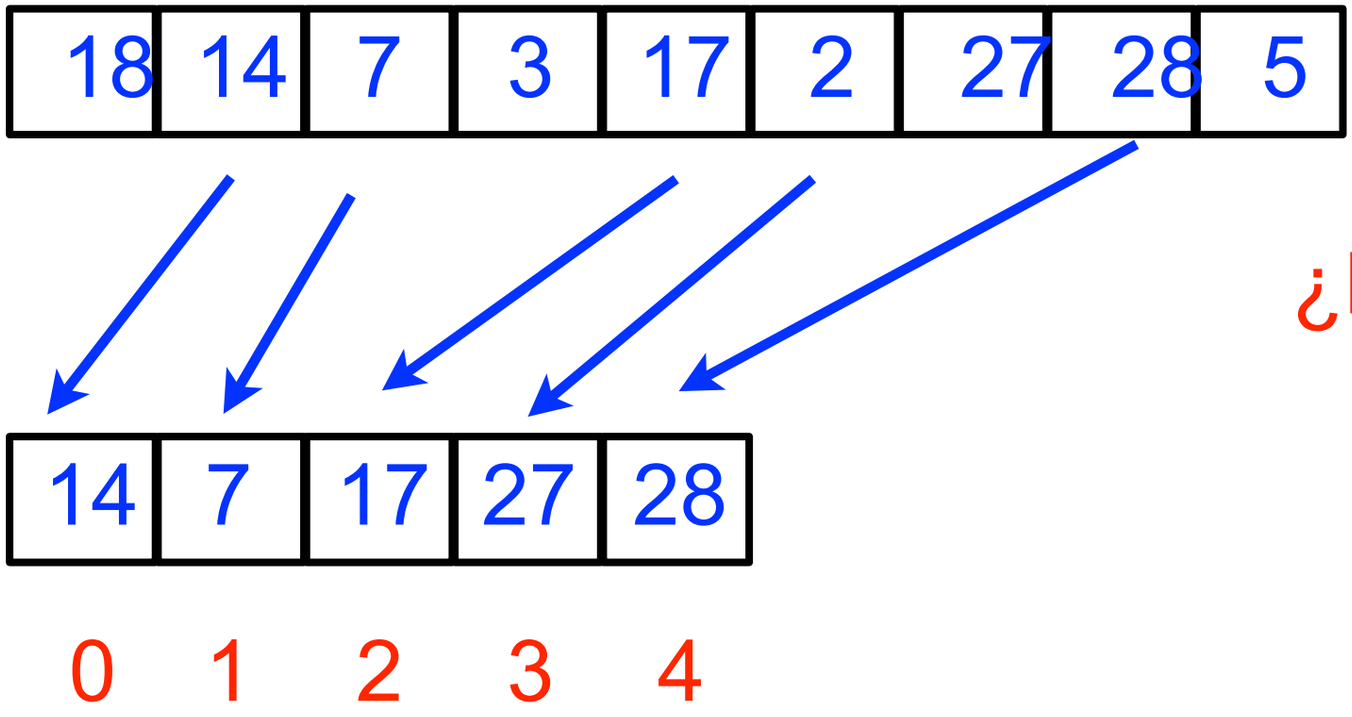


Block



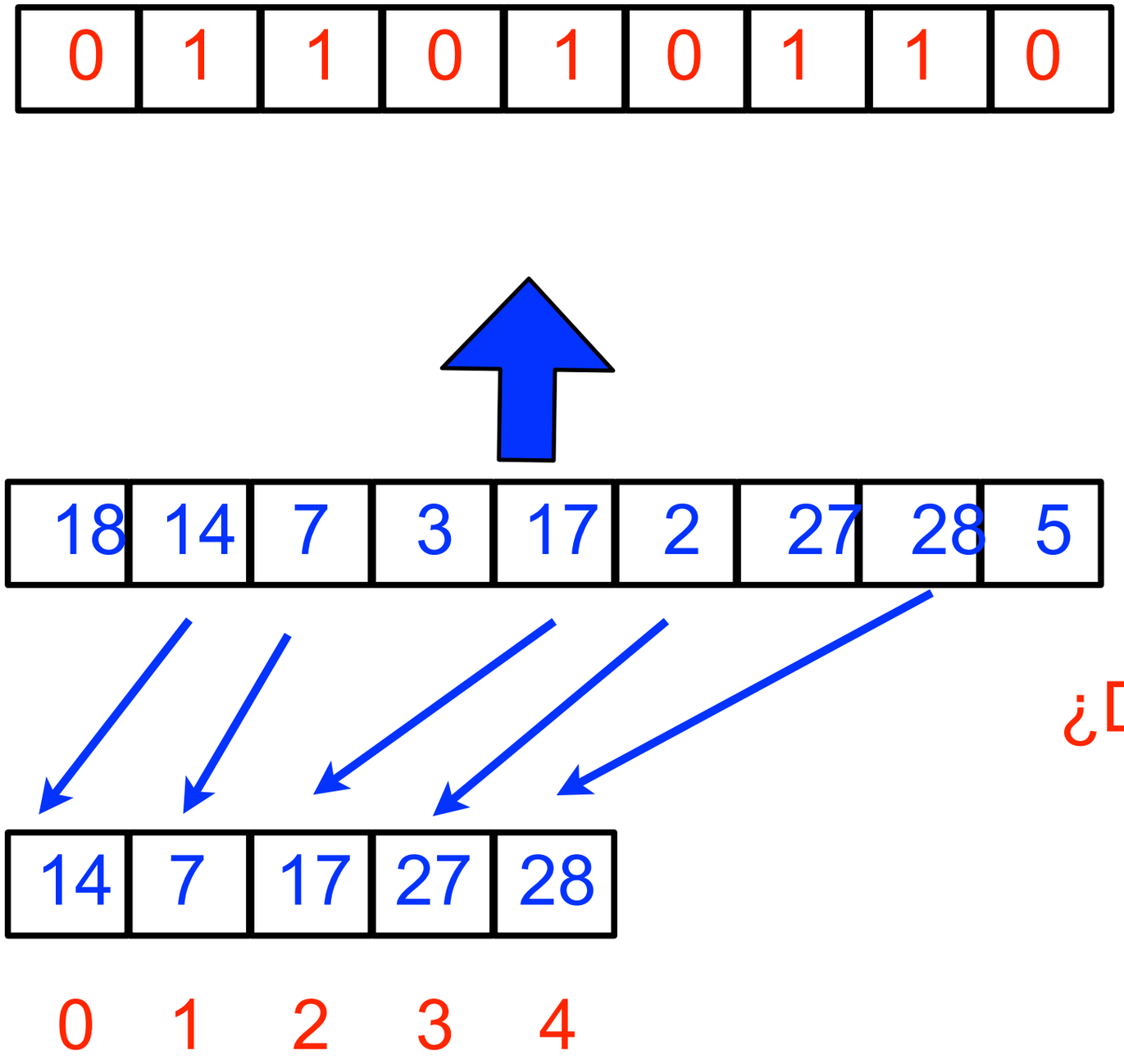
Compact

Quiero seleccionar sólo algunos



¿Dónde los escribo?

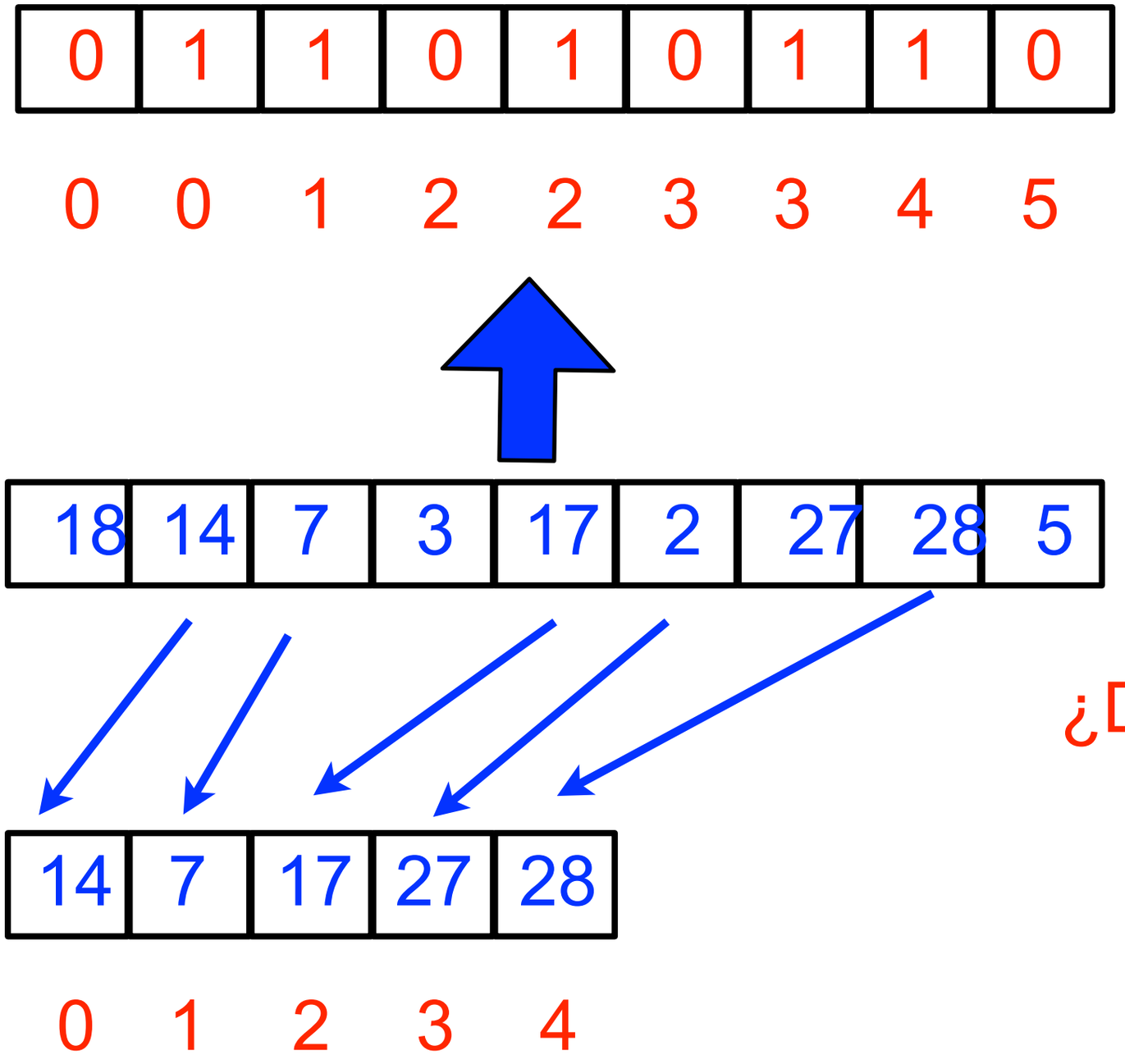
Compact



Quiero seleccionar sólo algunos

¿Dónde los escribo?

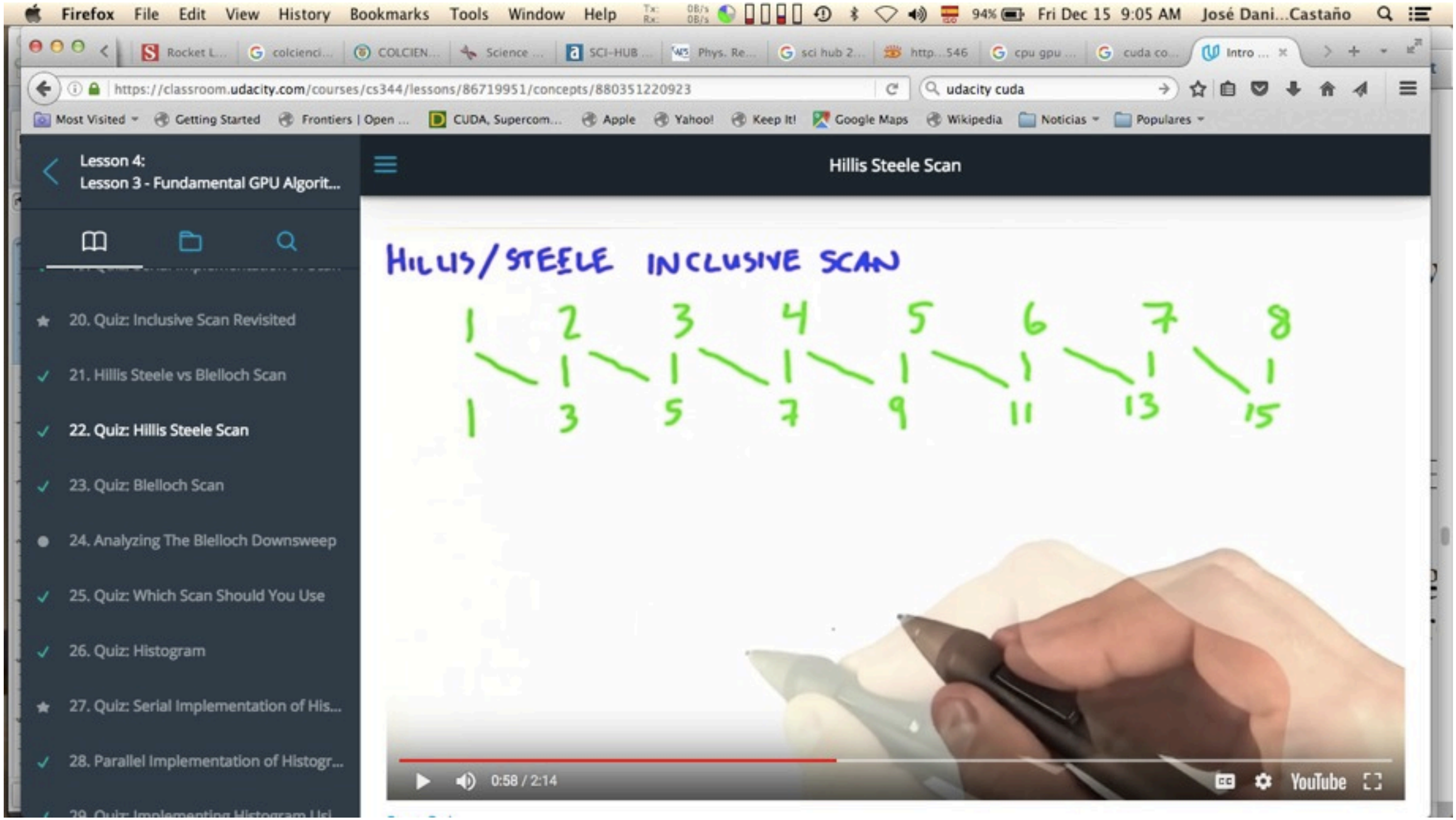
Compact



Quiero seleccionar sólo algunos

¿Dónde los escribo?

Curso de CUDA en Udacity



https://in.udacity.com/course/intro-to-parallel-programming--cs344

ssf+un

Gracias!

jdmunozc@unal.edu.co