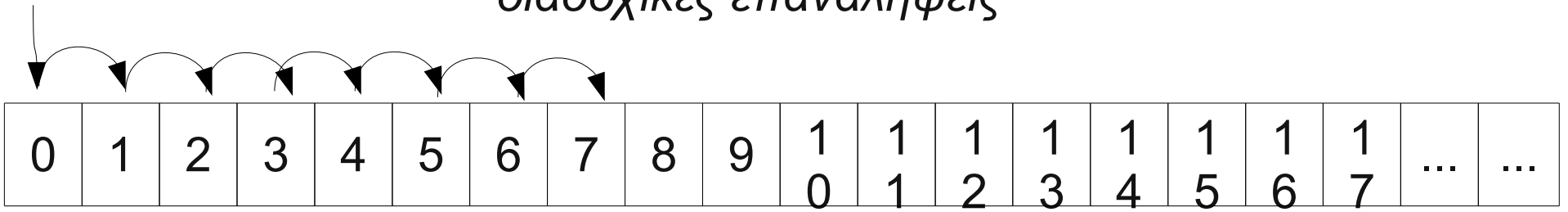
CPU

διαδοχικές επαναλήψεις

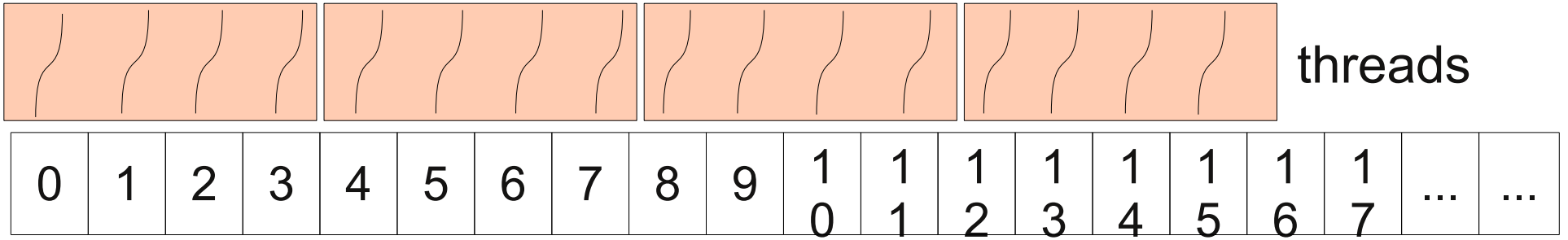


GPU

ταυτόχρονα

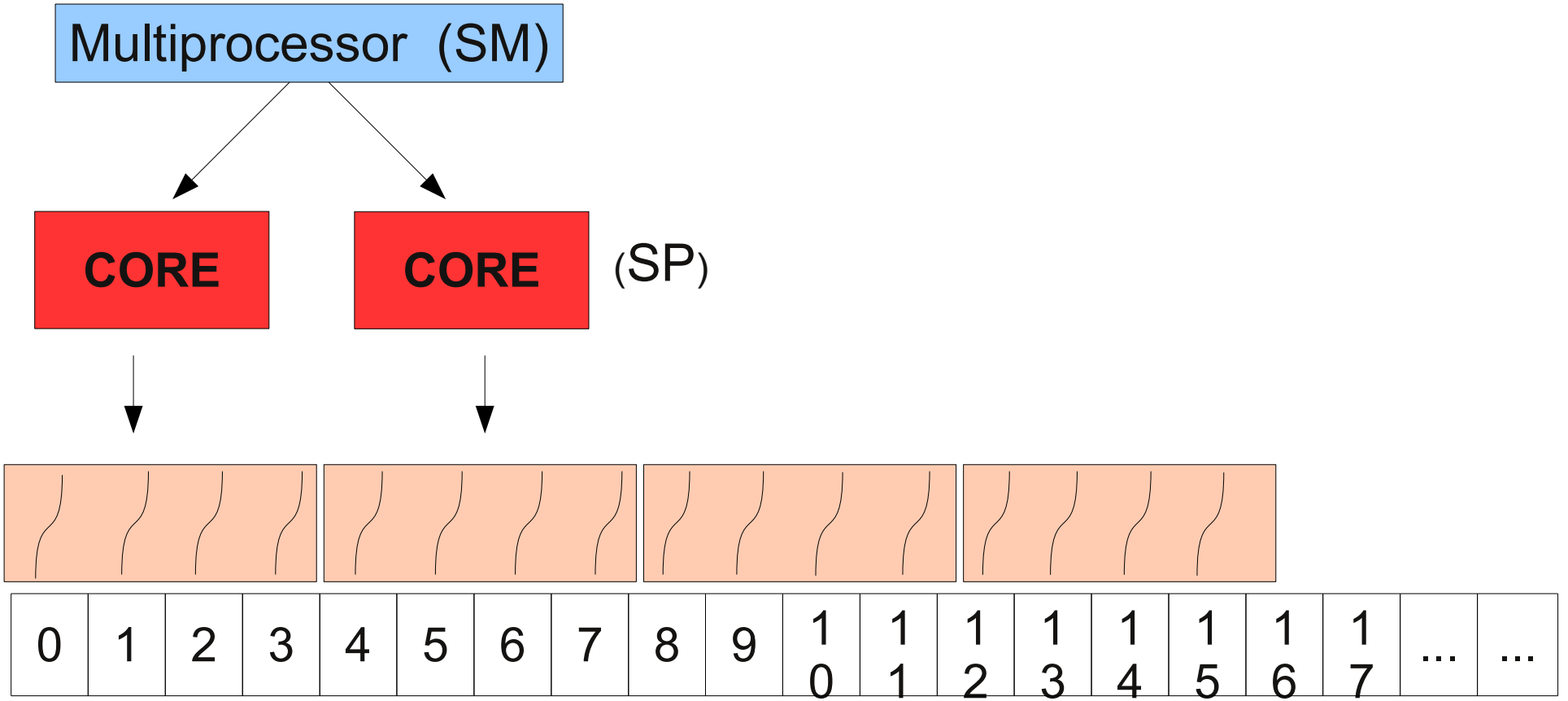
blocks

threads

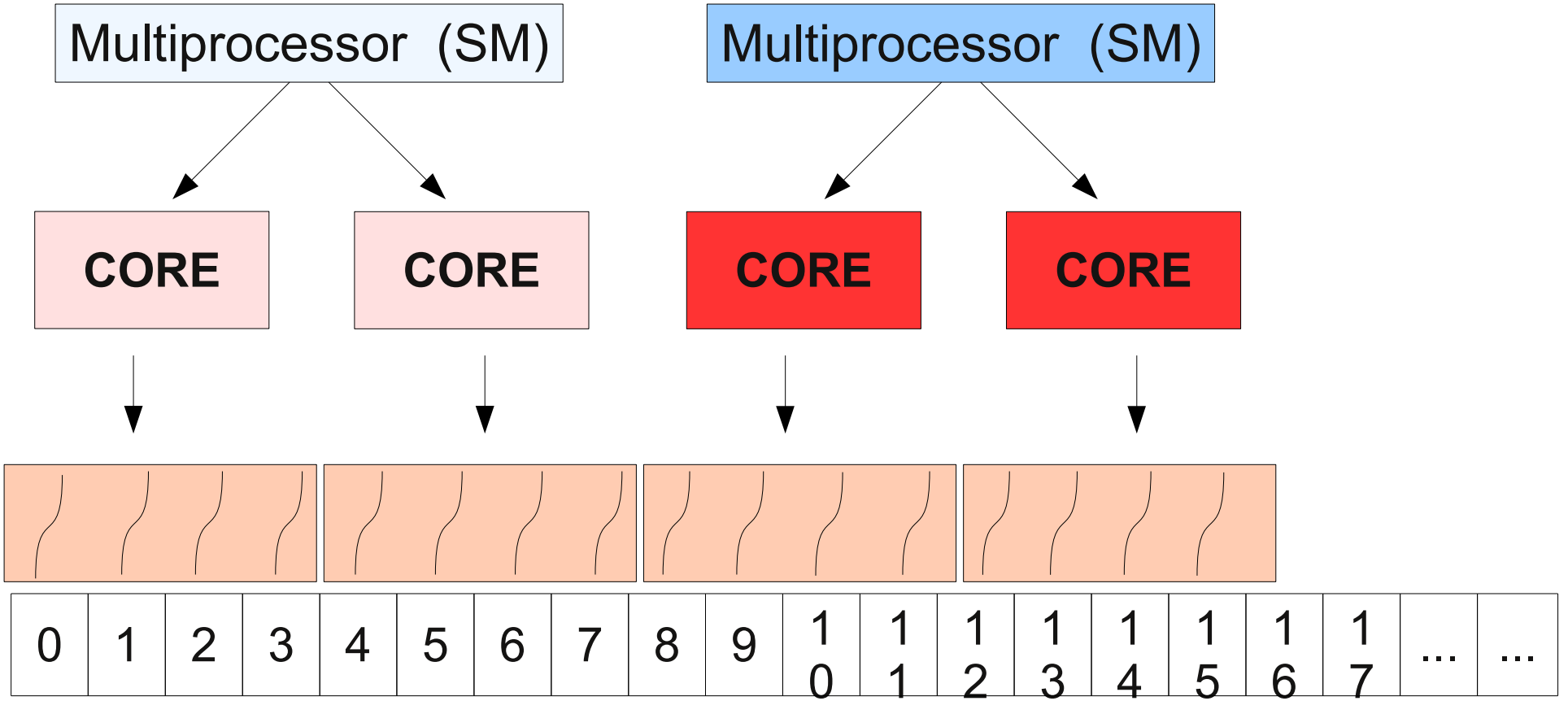


έχω 768 threads... και πίνακα 1024 στοιχείων (???)

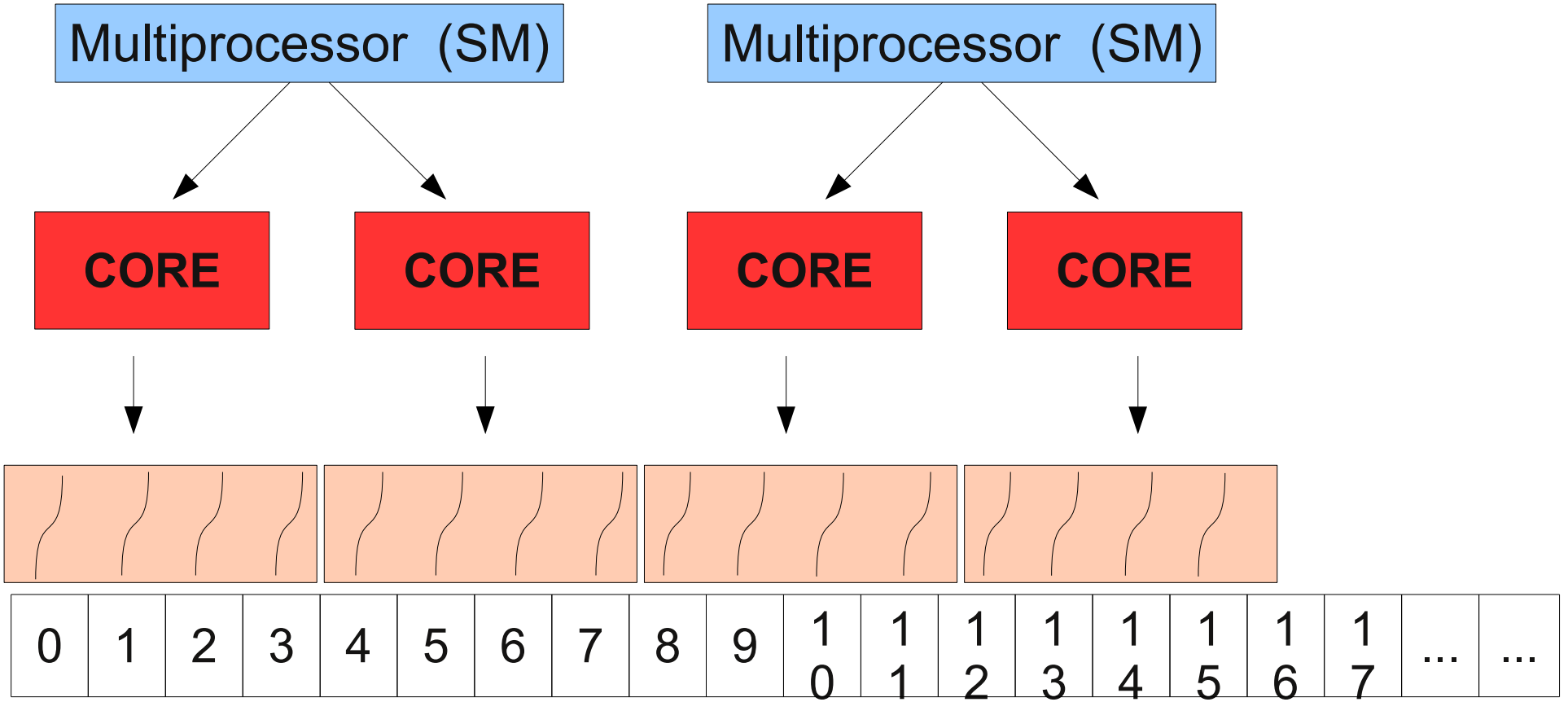
GPU


Multiprocessor (SM)
CORE
CORE
(SP)
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
...
...

GPU

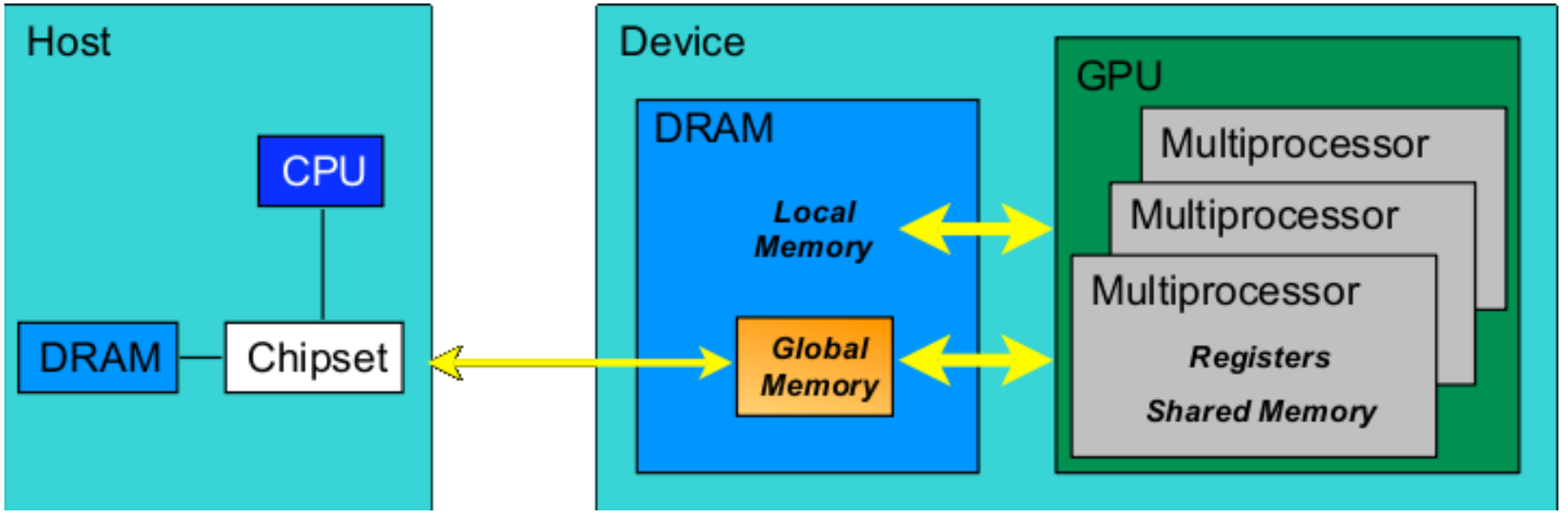

Multiprocessor (SM)
Multiprocessor (SM)
CORE
CORE
CORE
CORE
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
...
...

GPU (ή έχω περισσότερα - scalability)



χωρίς να επηρεάζεται καθόλου ο κώδικας

Χωριστές οντότητες



“Σύστημα” “Συσκευή”

Χωριστές μνήμες και προσβάσεις σε αυτές

Υλοποίηση ??? ⟶ kernel

Kernel

kernel <<<gridDM,blockDM>>> (....)

Συνάρτηση τύπου __global__

- τύπου void
- απαγορεύεται μεταβλητός αριθμός μεταβλητών
- δεν μπορεί να καλεί τον εαυτό του
- μπορεί να καλεί συναρτήσεις τύπου __device__
- όχι static μεταβλητές

Kernel

ορισμός:
__global__ void kernel (....)

κλήση:
kernel <<<gridsize,blocksize>>> (....)

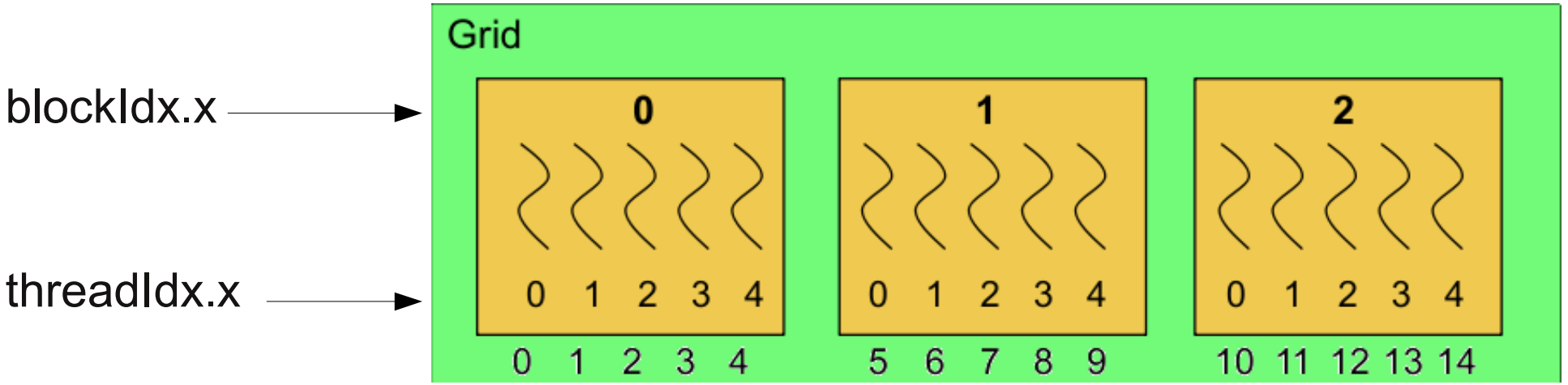
"execution configuration"

grid έως 2d

```
dim3 gridsize,  blocksize;
gridsize.x = 3 ;  gridsize.y = 6 ;
blocksize.x = 4 ;  blocksize.y = 7;  blocksize.z = 9;
```

block έως 3D

Kernel




```
__global__ void kernel (int* a, int* b, int* c) {
    idx = blockIdx.x * blockDim.x + threadIdx.x ;

    a[idx] = blockIdx.x;
    b[idx] = threadIdx.x;
    c[idx] = blockIdx.x * blockDim.x +threadIdx.x ;
}

kernel  <<<3,5>>>  (....);
```

επικοινωνία Device - Host

- cudaMalloc(void ** pointer, size_t nbytes)
- cudaMemset(void * pointer, int value, size_t count)
- cudaFree(void* pointer)

```
int nbytes = n*sizeof(int);
int *dev_a = 0;
cudaMalloc( (void**)&dev_a, nbytes );
cudaMemset( dev_a, 0, nbytes);
cudaFree(a_d);
```

cudaMemcpy(void *dst, void *src, size_t nbytes, enum cudaMemcpyKind direction);

- cudaMemcpyKind:
  - cudaMemcpyHostToDevice
  - cudaMemcpyDeviceToHost
  - cudaMemcpyDeviceToDevice

--παραδείγματα κώδικα--

occupancy calculator

Συγχρονισμοί:

Η ροή της εκτέλεσης στο σύστημα:

- Οι kernels εκτελούνται ασύγχρονα
- Η cudaMemcpy() εκτελείται “σύγχρονα”
- cudaThreadSynchronize()

Η ροή της εκτέλεσης στη συσκευή:

- sync όλου του grid δεν γίνεται! (αλλά πως;)
- sync όλου του block γίνεται: __syncthreads();

www.nvidia.com/cuda -> developers -> downloads

εκπαιδευτικό υλικό:
http://developer.nvidia.com/object/cuda_training.html

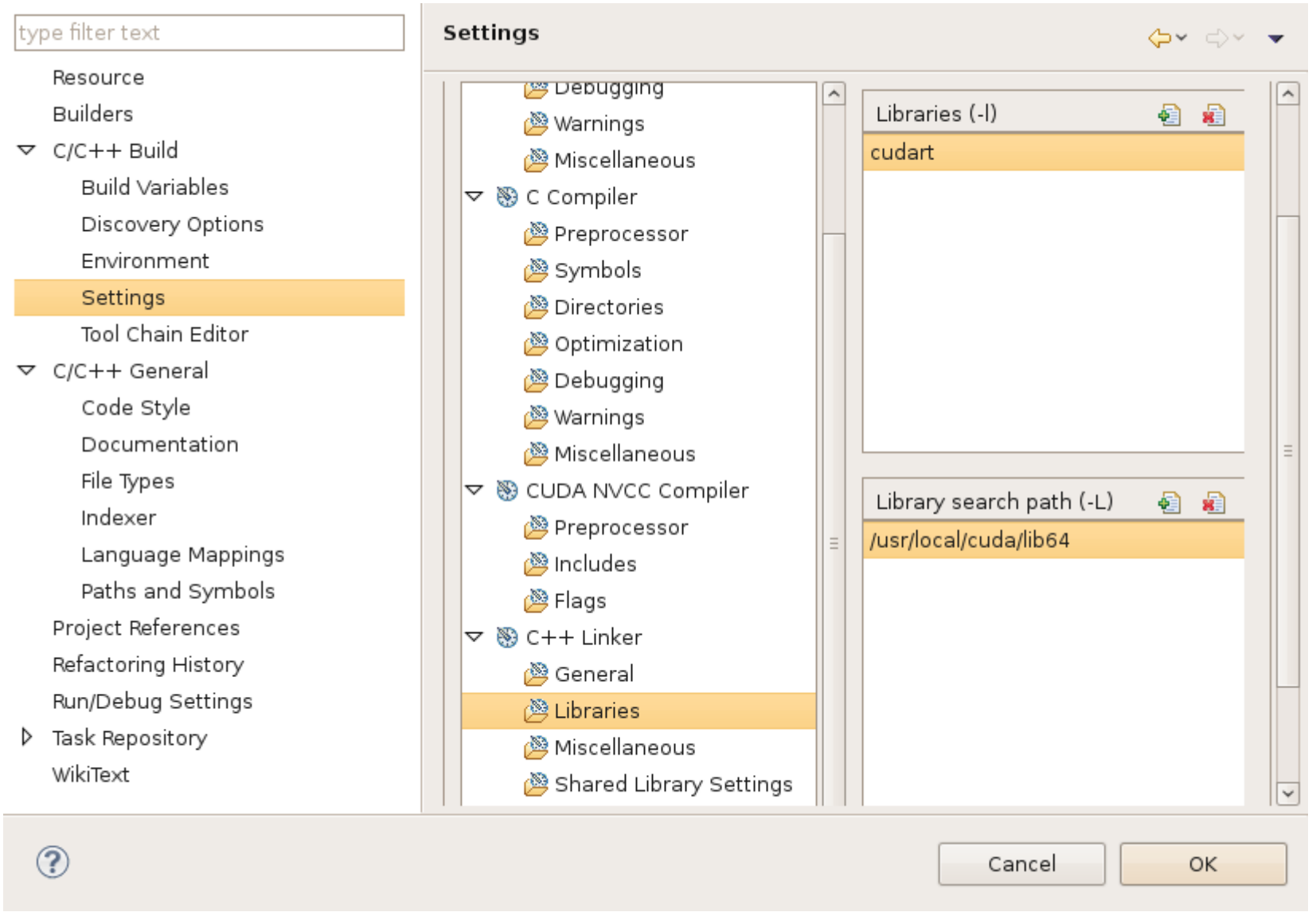
forums:
http://forums.nvidia.com

http://www.eclipse.org/ -> downloads -> eclipse IDE for C/C++ dev.....

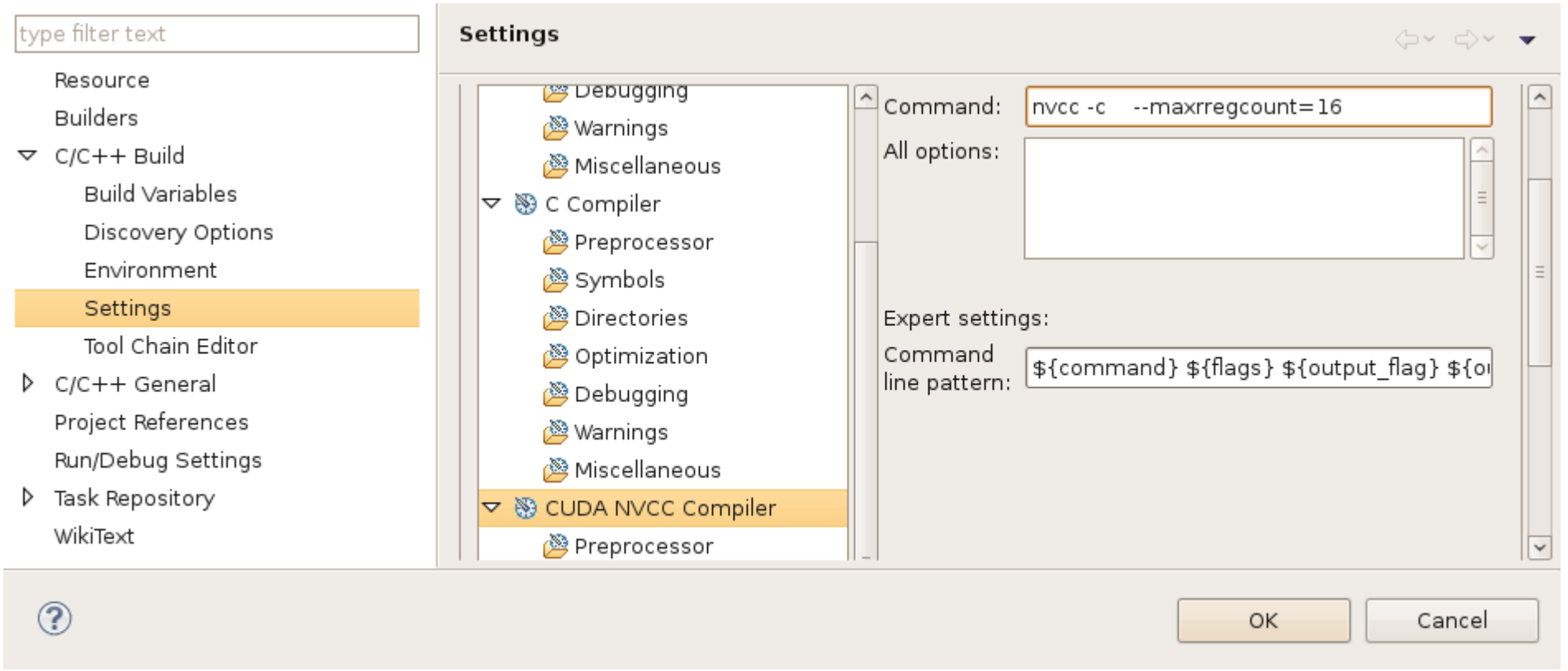
cuda plugin για το eclipse:
http://www.ai3.uni-bayreuth.de/ -> software -> eclipse toolchain for compiling CUDA & QT

προσθήκη της cudart library στο eclipse....

project -> properties -> ...



Περιορισμός των regs/thread στο eclipse



εδώ μπαίνουν και όποιες ρυθμίσεις του compiler