Basic elements of a CUDA program

Basic steps of a CUDA program

- Initialization of data from CPU
- transfer data from CPU context to GPU context

Kernel launch with needed grid/block size

- Transfer results back to CPU context from GPU context
- Reclaim the memory from both CPU and GPU

Elements of a CUDA program

- Host code (main function)

Code that is going to run in CPU

- Device code

Code that is going to run in GPU

return type

function name

argument list

```
int   hello_world (int x, float y, char * name)
```

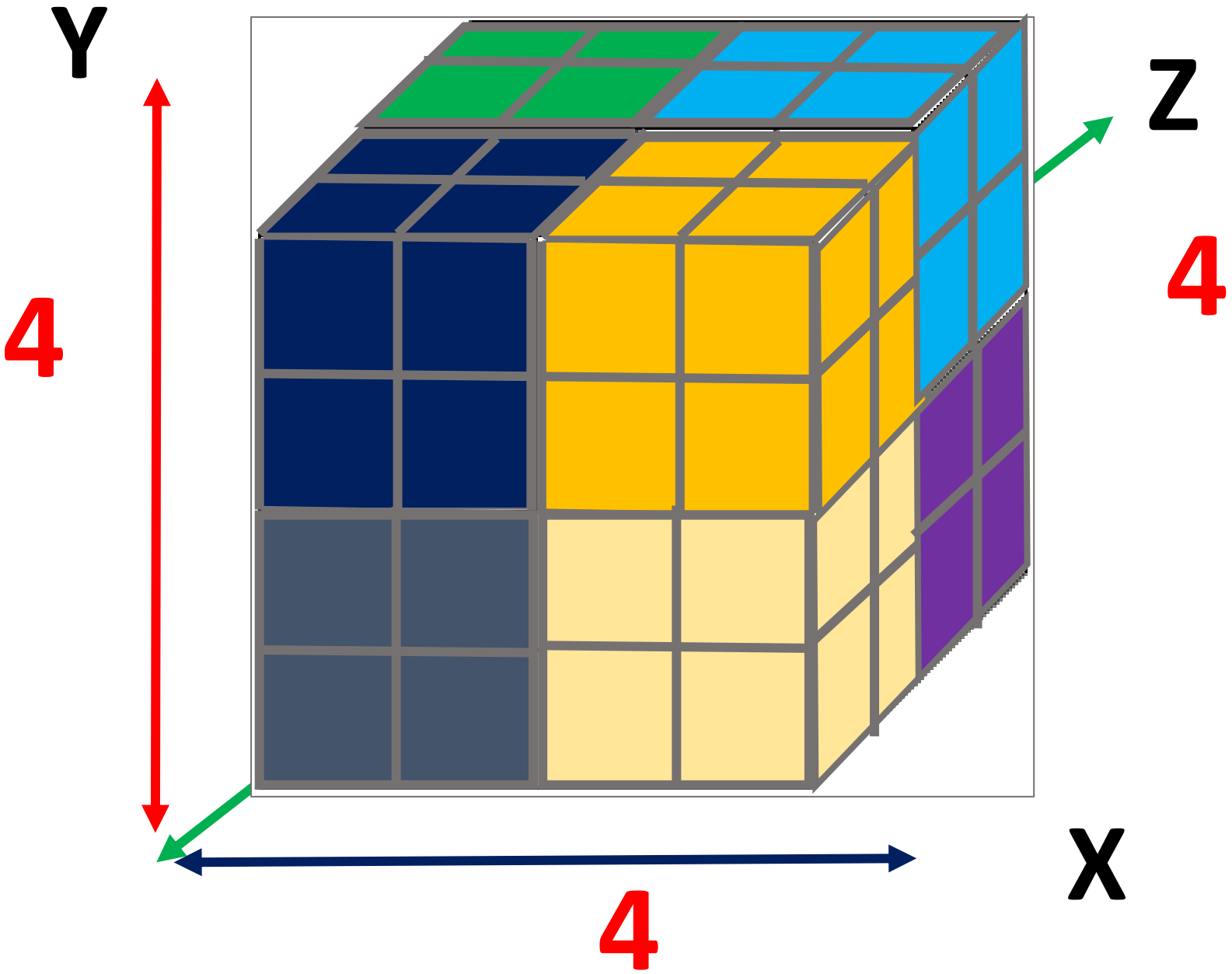
```
__global__   void hello_cuda (int x - - - - - -)
```

Grid

Grid is a collection of all the threads launch for a kernel

Block

Threads in a grid is organized in to groups called thread blocks


Y
4
Z
4
X
4

Kernel_name **<<<**

number_of_blocks,

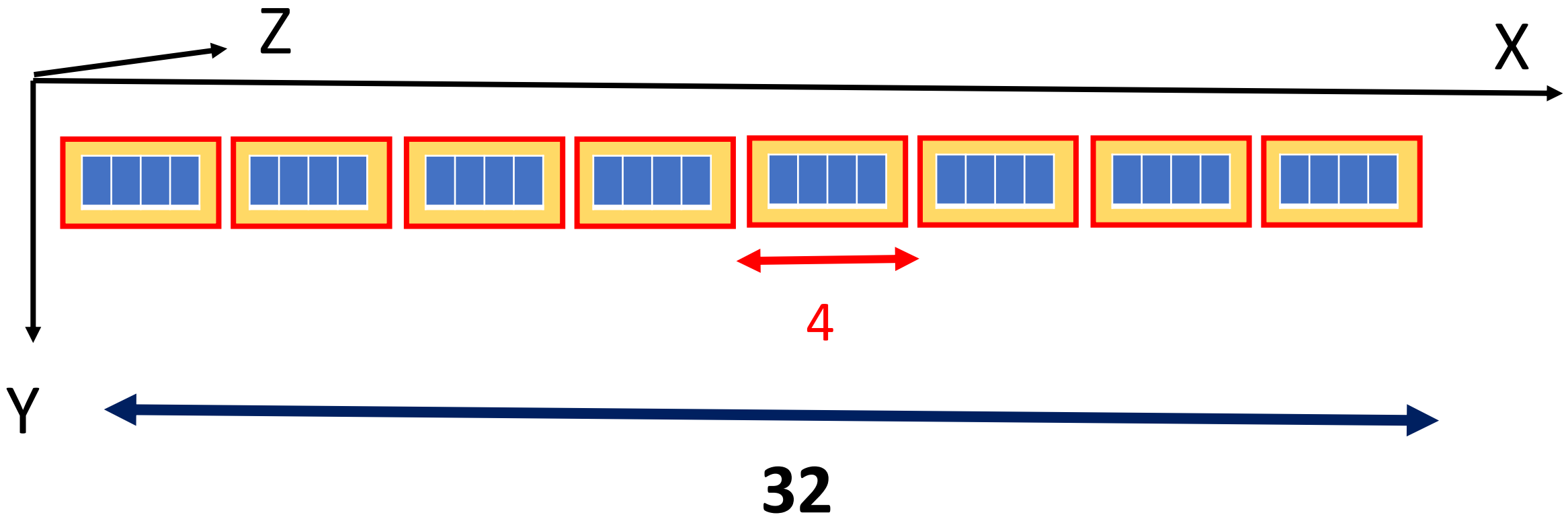
thread_per_block **>>>** (*arguments*)

dim3 variable_name (X, Y, Z)

variable_name.x

variable_name.y

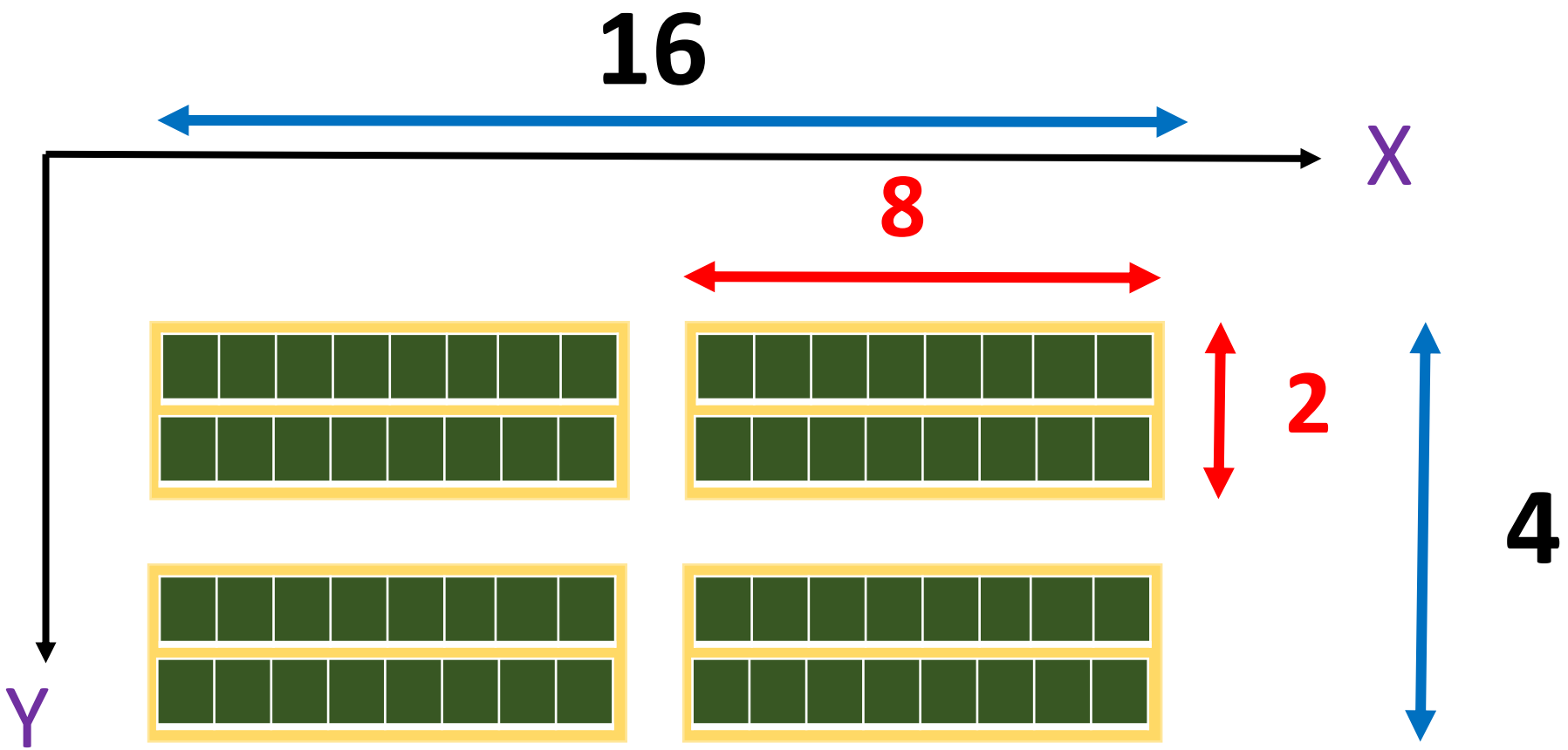
variable_name.z




```
dim3  block( 4, 1, 1)

dim3  grid( 8, 1, 1)

Kernel_name << grid, block >>>()
```

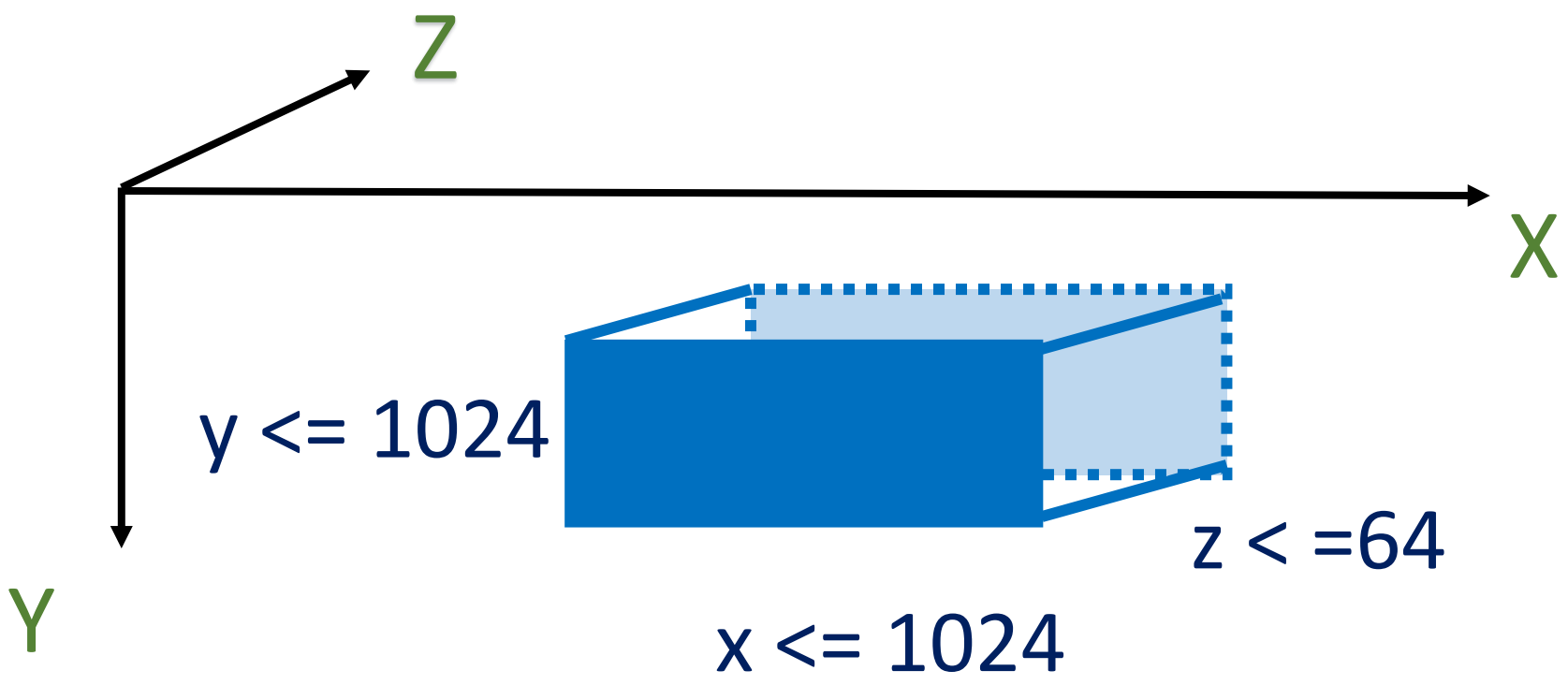


dim3 **block**(8, 2, 1)

dim3 **grid**(2, 2, 1)

Kernel_name << grid, block >>>()

Limitation for block size



$$x * y * z <= 1024$$

Limitation for number of thread block in each dimension