

**KU LEUVEN**



FACULTEIT  
INGENIEURSWETENSCHAPPEN

---

# Sync vs async execution

---

## Comparison

---

Benjamin Rübenkamp  
R0793577  
MEIsw  
Master industrieel ingenieur Elektronica-ICT  
Academiejaar 2023-2024

**\*\* There are no code examples in this text, please look in github repository under session5: github link \*\***

### **Introduction:**

This exercise consists of 2 parts where we see what impact async execution has on execution time. The exercise consists of making 4 arrays, filling them up with random values, and using the reduction kernel from session 2 to calculate the sum, product, min and max of the respective arrays. Both implementations do 1k iterations and only over the last 100 is the mean execution time calculated.

### **Part 1 (code.cu)**

For the first part of the exercise we start to time at the start of the program. All memory is allocated and arrays are initialised one by one. Then the 4 kernels are called one after another (after `cudaDeviceSynchronize()` to make everything synchronous) Finally the mean execution time is printed.

For 4 elements the mean execution time is 0,83ms, and for 10k elements it is 1,37ms.

### **Part 2 (code\_part 2.cu)**

This time memory copying and array generation happens async. While the GPU is executing a kernel, the cpu spends it's time generating the array for the next kernel. This leads to overall faster execution times.

For 4 elements the mean execution time is 0,55ms, and for 10k elements it is 1,28ms.