Lokman ABBAS TURKI

11. Reduction Exercise : The greatest value of an array

Executing `biggest_k<<<numBlocks, threadsPerBlock>>>(int *In, int *Out, int N)`, we want to find the largest element `Out` of an array `In` of `N` unsorted positive integers. To further simplify the exercise, the collaboration between threads is only implemented on the shared memory (the collaboration not involving registers) which must have a static allocation, for example :

`__shared__ int InSh[64];`

We also assume that the compilation is done with the options `-arch=compute_60 -code=sm_Yy where Yy>60`.

The student can use the same definition of kernels for several questions provided they explain how the definition is generic and has no negative impact on the speed of the calculation.

1. Define the kernel `biggest_k` that uses the maximum number of threads launched to find the maximum of the array `In` when executing
   a) `biggest_k<<<1, 1>>>(In, Out, 32);`
   b) `biggest_k<<<1, 2>>>(In, Out, 32);`
   c) `biggest_k<<<1, 4>>>(In, Out, 32);`
   d) `biggest_k<<<1, 64>>>(In, Out, 128);`

2. Unlike the kernel definition associated with question 1.d, why the one associated with `biggest_k<<<1, 32>>>(In, Out, 64);` or with one of the questions 1.a, 1.b, 1.c does not necessarily use `__syncthreads();` ?

3. Based on question 1.d, define `biggest_k` to find the maximum of the array `In` when executing `biggest_k<<<1, 64>>>(In, Out, 121);`

4. `atomicMax(int* address, int val);` is an atomic function that allows you to do the maximum between `val` and `address[0]` then returns the result in `address`. Based on question 1.d, define `biggest_k` to find the maximum of the array `In` when executing
   a) `biggest_k<<<2, 64>>>(In, Out, 256);`
   b) `biggest_k<<<(N+127)/128, 128>>>(In, Out, N);` with `N`> 256