Theory questions - ingara

Question 1)
__global__ functions are called kernels, and are the functions that the host will call to execute code on the CUDA device. __device__ functions are also executed on the CUDA device, but are not callable by the host. Rather, any function executing on the device (such as a kernel or another __device__ function) can call them.

Question 2)
Parallelizing using CUDA over MPI is advantageous if the computation you are performing is efficient on GPU hardware, typically floating point operations. GPUs have a much higher count of floating point operation hardware than any CPU of the same generation, with the number of cores ranging in the thousands to the tens of thousands for modern GPUs, compared to CPUs which range from single digit to double digit, maybe triple digit in modern extreme high-end CPUs. Each of these cores will likely have one or more FPUs, but these FPUs must share the memory buses, instruction pipelines and caches with non-floating point operations. Meanwhile, the entire GPU can be dedicated to such operations, invariably leading to greater performance (unless your CUDA program is really bad).

Question 3)
Cooperative groups gives the programmer greater flexibility in defining which threads have the ability to synchronize their work than before they were introduced. Before, __synchtreads() was the only mechanism, and it will perform synchronization on all threads of a threadblock without the possibility of dividing it into subsets, and cannot perform synchronization across threadblocks. Cooperative groups can do both of these tasks. This makes programming easier, and can also make programs more readable and explicit since it can be made clear which threads are related to each other. A limitation is that they are only available in CUDA versions >= 9, meaning that they can't be used on GPUs older than CC 3.0 (https://developer.nvidia.com/blog/cooperative-groups/). Another limitation is that all threads in a group must make calls to any collective operation introduced in the code path.

Question 4)
I achieved a theoretical occupancy of 1.0, so it cannot be increased. However, this does not necessarily mean that it cannot be improved, as a high thread occupancy is not always desirable due to the possibility of resource contention. However, the experimentation needed to find the optimal block and grid sizes is outside the scope of this exercise, in my opinion.