CSCI-GA.3033-012

Graphics Processing Units (GPUs): Architecture and Programming

Spring 2012 – May 10th, 2012

NAME: **ID:**

- This exam contains 5 questions with a total of 40 points.
- The exam is open book/notes but no electronic devices.
- If you have to make assumptions to continue solving a problem, state your assumptions clearly.
- You answer on the question sheet. You can use extra white papers if you want.

1. [5 points]We have a 2D array with X rows and Y columns. For best performance, the array needs to be memory aligned on 64 byte boundary (as we have seen in pitch). Derive a formula to calculate percentage of wasted memory (the padding) in terms of X and Y, assuming that a single array element is one byte. Assume that the array is NOT already aligned (i.e. ignore the special case where no padding is necessary).

2. [5 points] Convert **126.0** to IEEE 754 Standard floating point. Show all steps.

3. [10 points] Write a kernel that finds the location of all occurrences of zero in array of integers INPUT, and writes the locations of the zero elements to integer array z_locations. For example if INPUT is {0, 0, 4, 0, 8, 9} then z_locations will contain {0,1,3}.

Assumptions: both INPUT and z_locations are array of integers. Each array contains NUM elements. All elements in z_locations are already initialized to -1, so do not include initialization phase for z_locations. INPUT is already pre-loaded with the elements. You just need to write the kernel function and not a full program. Also assume that sizes of blocks and grid have already been set. Both INPUT and z_locations are already in the GPU global memory.

After finishing your code, indicate all the optimizations you did in your code, if any.

4. [4 points] In FERMI memory hierarchy we have 64KB that can be configured as 48K shared memory and 16KB L1 cache or 48KB L1 cache and 16KB shared memory. Indicate when will you use the first configuration and when will you use the second configuration.

5. [6 points] Assume we have M total threads, each of which need some data from the global memory of the GPU. Those threads can be grouped into X blocks with Y threads each (i.e. X*Y=M). Keeping the total number of threads fixed, discuss the effect of increasing X (and decreasing Y to keep M fixed) or increasing Y on bandwidth requirement. Assume the GPU can accommodate M total threads per SM and only 1 block per SM, and the total number of SMs is M (i.e. a maximum of M*M threads can exist in the whole GPU at the same time). Justify your answer.