EEP524
Homework 1
Jaidon Lybbert

Questions

1)

The expression for mapping the thread/block indices to data index is
C. **i=blockIdx.x*blockDim.x+threadIdx.x**

2)

The expression to map the thread/block indices to the data index of the first element for calculating two adjacent output elements of a vector is addition is

A. i=blockIdx.x*blockDim.x + threadIdx.x +2;

3)

The expression to map the thread/block indices to the data index of the first element is
D. i=blockIdx.x*blockDim.x*2 + threadIdx.x

4)

The number of threads in the grid is
C. 8192

Cuda Kernels

Printf

4) Questions

a) What do you observe about the results?

   The order of print statements is sequential for threads within the same block, but the blocks are not printed in order, rather it is random which block is printed first.

b) Does the behavior of block and thread indices seem to differ?

The behavior does differ. The threads are ordered per block, but the blocks are not ordered per grid.

c) What does this imply about computations you perform in the kernels?

Some form of deterministic scheduling exists such that kernels executing on threads in the same block return in sequence, but nothing can be predicted about kernels executing on threads of different blocks.

Host-Side Result Verification

c) At what tolerance values do verification results fail?

I wasn't able to discern any differences between the CPU and GPU results. I populated the inputs with randomly generated values, then calculated the Mean Squared Error between the CPU and GPU results and found no error.

d) Which of the 3 kernels exhibits the most severe result disagreements between CPU and GPU versions? Why?

I didn't see any error for any of the 3 kernels. Since we are only doing addition with existing 32-bit floating point values in memory, and all the input values I used were of the same magnitude, I wouldn't expect to see many rounding errors.