Homework Assignment #3

- DUE 5PM Wednesday, March 4
- Objective:
  - Part I: Develop coding experience in CUDA using tiling to shared memory to improve performance
  - Part II: Experiment with the performance impact of various code optimizations. Share results with rest of class to develop collective knowledge.
- Turning in assignment:
  - Use the "handin" program on the CADE machines
  - Use the following command:

    "handin cs6963 lab2 <prob1file> <prob2file>"
  - The file <prob1file> should be a gzipped tar file of the CUDA program and output for Problem 1
  - The file <prob2file> should be a gzipped tar file of the CUDA program and output for Problem 2

Homework Assignment #3

Problem 1: Tiling for Shared Memory

Consider the following program fragment, which is a Jacobi relaxation algorithm:

```
…

#define n 64

float a[n][n][n], b[n][n][n];

 for (i=1; i<n-1; i++)

   for (j=1; j<n-1; j++)

     for (k=1; k<n-1; k++) {

        a[i][j][k]=0.8*(b[i-1][j][k]+b[i+1][j][k]+b[i][j-1][k]  +

                             b[i][j+1][k]+b[i][j][k-1]+b[i][j][k+1]);

     }
```

Homework Assignment #3

Problem 1, cont.: This type of computation accesses the "nearest neighbors" of a data point, applying a function to the neighbors to compute the value at the current point. Such computations are sometimes referred to as stencils.

The access pattern has reuse on array B in 3 dimensions. Your assignment is to write a CUDA version of Jacobi that applies tiling to exploit locality in shared memory for the 3 dimensions of B.

You will be provided with a sequential CPU implementation of the code, which extends the above loop nest to include initialization of B and final output. Please use the timing and comparison to the CPU results found in the matrix multiply example code.

Homework Assignment #3

Problem 2: Select one of the following questions below. Write a CUDA program that illustrates the "optimization benefit" (OB) or "performance cliff" (PC) in the example. These codes will be shared with the rest of the class. Also provide a brief (a few sentences) description of what is happening as a comment inside the code.

a. [PC] Show an example code where you fill up the register file due to too many threads. You should have two versions of the code, one where the number of threads is within the range of registers, and one where the register capacity is exceeded.

b. [OB] Show the performance impact of unrolling an innermost loop in a nest. See how far you can push it before you run into the problems of a. above.

c. [OB/PC] Explore when the compiler decides to put array variables that are local to the device function in registers. What access patterns lead to the compiler using a register vs. using local memory.

d. [OB/PC] Show the performance advantage of constant memory when the data is cached, and what happens to performance when the data exceeds the cache capacity and locality is not realized.

Homework Assignment #3

Problem 2, cont.:

e. [OB] Show the performance impact of control flow versus no control flow. For example, use the trick from slide #13 of Lecture 9 and compare against testing for divide by 0.

f. [PC] Demonstrate the performance impact of parallel memory access (no bank conflicts) in shared memory. For example, implement a reduction computation like in Lecture 9 in shared memory, with one version demonstrating bank conflicts and the other without.

g. [OB] Show the performance impact of global memory coalescing by experimenting with different data and computation partitions in the matrix addition example from lab1.