ME759
High Performance Computing for Engineering Applications
Assignment 8
Parallel Prefix Scan
Date Assigned: October 28, 2013
Date Due: November 4, 2013 – 11:59 PM

The goals of this assignment are as follows:
A) Implement a tiled parallel solution of the exclusive scan operation discussed in class so that one can perform exclusive scan on more than 2048 elements. The number of elements however is going to be a power of two (see below).
B) Implement a work efficient algorithm to get either $O(n)$ or $O(n\log(n))$ performance.
For extra credit:
C) Implement your solution such that you can handle non-power-of-2 array sizes.
D) Implement your solution such that it minimizes shared memory bank conflicts.

To meet the goals of this assignment:
1) Edit the source files **`scan_largearray.cu`** to implement the necessary kernel functions. Use a tiled implementation so that computation can be performed on more than 2048 elements. Feel free to edit this file as you see fit. You might want to add, delete, or modify code; you might choose to ignore this altogether and start from scratch.

2) The modes of operation for the application are as follows:

a) No arguments: Randomly generate input data and compare against the host's result.

b) One argument: Randomly generate input data and write the result to a file, the name of which is specified by the first argument.

c) Two arguments: Read the first argument which indicates the size of the array (a power of 2), randomly generate input data and write the input data to the second argument (for generating random input data).

d) Three arguments: Read the first argument which indicate the size of the array, then input data from the file name specified by 2nd argument and write the output to a file name specified by the 3rd argument.

Note that if you wish to use the output of one run of the application as an input, you must delete the first line in the output file, which displays the accuracy of the values within the file. The value is not relevant for this application.

3) Please use the following steps for your performance analysis and report:

i) Near the top of the file **`scan_largearray.cu`**, set `#define DEFAULT_NUM_ELEMENTS` to 16777216 (which is 2^{24}). Set `#define MAX_RAND` to 2.

ii) Include in your assignment report and also post on the ME759 Forum the performance results when running the executable without arguments on the CPU and GPU. Note that you are expected to report inclusive times in Release build mode.

iii) Using the NVIDIA profiler, generate and post on the forum and include in your report the **`nvvp`** snapshot that shows how the execution time was spent by your program. Comment in your report on the amount of time spent in memory transactions and the bandwidth that your program displays.

iv) Describe which of the four goals for this assignment (mentioned in the beginning) were achieved. How did you achieve them? Did you use any tricks to optimize for best performance? Please briefly discuss them if any.

Upload your solution at Learn@UW.

Grading:
Your submission will be graded according with the following scheme:

Demo/knowledge: 25%

- Produces correct result output file for command-line inputs

Functionality: 40%

- Correct identification and expression of parallelism in the problem (10%)
- Blocked Implementation (15%)
- Work Efficient Algorithm (15%)
- Handling of non-power-of-2 array sizes (15%) (BONUS)
- Implementation minimizes shared memory bank conflicts (15%) (BONUS)

Report: 35%

- Answer to provided question

NOTES:

- The student with the fastest code will be mentioned in the "Assignment Champion[s]" forum posting.
- Further information on the parallel scan operation:
  http://graphics.idav.ucdavis.edu/publications/print_pub?pub_id=915