ECEC 414: High-Performance Computing
CUDA Programming Assignment 2

Prof. Naga Kandasamy, ECE Department, Drexel University

November 6, 2014

The lab is due on November 23, 2014. You may work on the problems in teams of up to two people.

Matrix-Vector Multiplication. You will multiply a dense $n \times n$ matrix A with an $n \times 1$ vector x to yield the $n \times 1$ result vector y. The serial algorithm is shown below.

```
1: procedure VEC_MAT_MULT(A, x, y)
2: int i, j;
3: for i := 0 to n − 1 do
4:     y[i] := 0;
5:     for j := 0 to n − 1 do
6:         y[i] := y[i] + A[i, j] × x[j];
7:     end for
8: end for
```

Edit the `vec_mat_mult_on_device()` function in `vec_mat_mult.cu` and the corresponding kernel function in `vec_mat_mult_kernel.cu` to complete the functionality of the vector-matrix multiplication on the GPU. The CUDA source files for this question are available on BBLearn as a zip file. Your program should accept no arguments. The application will create a randomly initialized matrix and a vector to multiply. After the GPU-based multiplication kernel is invoked, it will then compute the correct solution using the CPU and compare that solution with the GPU-computed solutions. If the solutions match within a certain tolerance, the application will print out "Test PASSED" to the screen before exiting.

Upload all of the files needed to run your code as a single zip file on BBLearn called `cuda_lab_2.zip`. This question will be graded on the following parameters:

- Make judicial use of the GPU shared memory to obtain the best speedup that you can over the CPU version, for matrix sizes of 4096×4096 and 8192×8192. When timing the GPU kernel, you may ignore the CPU-GPU data transfer overhead.
- Include a brief report describing how you designed your kernel (use code or pseudocode to clarify the discussion) and the amount of speedup obtained over the serial version for both GPU-based versions.