University of Arkansas
Department of Computer Science & Computer Engineering
CSCE4643/5693 – GPU Programming

Lab 5: Reduction

Due Date: March 12 (Thursday) at midnight 11:59 pm
Required for ALL students

1. Objective

The objective of this lab is to get you familiar with the parallel reduction algorithm.

2. Procedure

Step 1: Download the lab 5 materials from blackboard to your home folder at Karpinski computer cluster. Unzip it.

```
unzip lab5-reduction.assignment.zip
```

Step 2: Edit `kernel.cu` where indicated to implement the device kernel code for the parallel reduction algorithm, assuming that an input array of any size can be handled by your kernel. Each thread block will deal with a sub-array of 2×BLOCK_SIZE. You only have to produce the partial sums of each thread block for this part. We will sum up the partial results to the final result on the host.

Step 3: Compile and test your code.

```
make
./reduction                # Uses the default input size
./reduction <m>            # Uses an input with size m
```

Step 4: Answer the following questions in a new file named `answers.txt`:

- How many times does a single thread block synchronize to reduce its portion of the array to a single value?
- What is the minimum, maximum, and average number of "real" operations that a thread will perform? "Real" operations are those that directly contribute to the final reduction value.

Step 5: Submit your assignment. You should only submit the following files:

- `kernel.cu`
- `answers.txt`

Compress the files and name them after your last name like the following:

```
tar -cvf lab5_<your last name>.tar kernel.cu answers.txt
```

Submit the tar file in moodle.

3. Grading:

Your submission will be graded based on the following criteria.

- Functionality/knowledge: 90 points
  - Correct code and output results

- Correct usage of shared memory in the kernel to hide global memory access latencies
- Avoiding control flow divergences and shard memory bank conflicts
- Correct handling of boundary cases
- Checking return values of CUDA APIs

- Answers to questions: 10 points
    - Correct answer to questions in Step 4
    - Sufficient work is shown
    - Neatness and clarity