Theodore Jagodits / CS 677 / HW #2 Report /

“I pledge my honor that I have abided by the Stevens Honor system”

1.4

a) I start off with 256 threads, which equals to log2(256) = 8 thread synchronizations within my thread block. This is because I add the elements from global into the shared memory in the first computation, so I have only 256 elements to do the algorithm on, which halves every time until the stride >= 1. Every time it halves is when I must sync the threads.

b) I am assuming that storing the result into memory is also an operation.

The first operation is bringing in from global memory and putting into shared memory, then finishing the first for loop iteration, every thread has to do this. So the minimum number of operations for a thread are two memory operations from global, accessing 2 values from the global array and then 1 addition operation, and one run of the for loop. The for loop calculates the next stride and accesses and adds to the shared memory. This totals to 2 global accesses, 3 shared accesses, 2 additions.

The maximum amount for a thread is the first one with id 0. It has to the for loop 8 times and store the result in global.

Therefore the max is, 3 global memory accesses, 10 shared accesses, and 9 additions.

The average amount of operations is calculated by taking the threads at each step of the for loop and calculating how many are actually adding into the stride.

The average is, 3 global memory accesses, 4 shared accesses, and 3 additions. This is because the most additions occur in the first for loop.