## Problem 1

(a) How many times does your thread block synchronize to reduce the array of 512 elements to a single value? (4 points)

Load shared memory 1 time, each time, the value will be half, 9 times to make 512 to 1 value .

(b) 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. (6 point)

Maximum operation : thread 0 and thread 1 , 9 times

Minimum operation : thread 256 — thread 511, 1 time.

Total operations = 256 * 1 + 128 * 2 + 64 * 3 + … + 2 * 8 + 2 * 9 = 1022.

Average operation = 1022 / 512 = 2.

## Problem 3

The following scalar product code tests your understanding of the basic CUDA model. The code computes 1024 dot products, each of which is calculated from a pair of 256-element vectors. Assume that the code is executed on G80. Use the code to answer the following questions.

a) How many threads are there in total? (1 point)

1024 * 256 = 262144

b) How many threads are there in a Warp? (1 point)

32

c) How many threads are there in a Block? (1 point)

256

d) How many global memory loads and stores are done for each thread? (3 points)

Each thread loads 2 float from d_A and d_B to shared memory,

stores Each block store a float to d_C

e) How many accesses to shared memory are done for each block? (4 points)

256 + 510 = 766.

f) List the source code lines, if any, that cause shared memory bank conflicts. (4 points)

There is no bank conflicts.

g) How many iterations of the for loop (line 23) will have branch divergence? Show your derivation. (6 points)

When stride <= 16, the threads in wrap have different branch, there are 5 iterations of loop have branch divergence.

h) Identify an opportunity to significantly reduce the bandwidth requirement on the global memory. How would you achieve this? How many accesses can you eliminate? (8 points)

Global Memory Coalescing, and use more threads in each block.