Parallel Computing
Homework Assignment # 3
[Total 30 points]

1. [5] If we have two implementations of the same algorithm and we found that one implementation has higher speedup than the other. Does that always mean that the implementation with higher speedup will always have higher efficiency? Justify.

2. We have two different implementations of the same algorithm. The first implementation has one million instructions where half of them are floating point instructions, and the other half are integer instructions. The second implementation has two million instructions where all of them are integer instructions. Suppose a floating-point instruction takes 10 cycles while an integer instruction takes one cycle. We execute these two implementations on a 2GHz machine. Assume single SISD core.

a. [4] What is the MIPS of each implementation?
b. [4] What is the CPI of each implementation?
c. [4] What is the total execution time of each implementation?
d [4] State which implementation is better based on each of the following measurements: instruction count, MIPS, CPI, and execution time.

3. [9 points 3 points for each a, b, and c]
Suppose we have a quad-core processor. Each core is a pipelined core with no superscalar or hyperthreading capabilities. Each has its own L1 cache. There is a shared L2 cache (i.e. all cores can access the same L2 cache). The block size for all caches is 32 bytes in size. We have an array of 256 integers. We want to increment each element of that array by one. We want to do that by writing a multithreaded program. The elements of this array will be divided among the four cores. State which elements of that array will be assigned to which core to reduce the overhead of coherence, in each one of the following scenarios. Also, for each one, give one sentence justification:

a. Each L1 cache is 64KB of size and is 2-way set associative.
b. Each L1 cache is 32KB of size and is 4-way set associative.
c. Each L1 cache is 16KB and is 8-way set associate.

[Definition: x-way set associative cache means that the cache is divided into sets and each set can hold x blocks]