Parallel Computing
Homework Assignment 1
[30 points]

1. [4] In the global sum problem that we discussed in class, in lecture 1, if we assume that there is a variable called *my_rank* (local to each core) that gives each core a unique rank from 0 to p-1 (for p cores), devise an expression to calculate *my_first_i* and *my_last_i* assuming:

a. n is divisible by p and n > p.
b. n is not divisible by p.

2. [8] In superscalar processors, we increase the number of execution units to be able to execute several instructions at the same time. What modifications do we have to make for the fetch, decode, issue and commit phases?

3. [6] Suppose we have a *sixteen-core processor*. For each one of the following scenarios, indicate the maximum number of threads that can be executed at the same time and why.

a) Each core is superscalar but not hyperthreading
b) Each core is superscalar and four-way hyperthreading
c) Each core is neither superscalar nor hyperthreading (just pipelining)

4. [3] Superscalar capability makes the processor execute several instructions at the same time. This is called instruction-level-parallelism. We measure the performance of this hardware using IPC (Instructions per cycle). Is speculative execution needed for superscalar processors to work? Justify

5. [3] Suppose with have a core with eight execution units. What is the difference between having this core be superscalar and having this core be two-way hyperthreading? In both cases we have only eight execution units.

6. [6] Before multicore processors, that is, during the single core era, programs are getting faster with every new generation of processors without any effort from the programmer. This is due to two factors. What are they? And why did each factor make the performance higher?