Parallel Computing

Homework Assignment #1

[30 points]

1. [2] We saw in class that using this “tree-like” way of adding several numbers from different cores is better than making all the cores send their number of a master-core to add it. Why is that?

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 to get the best performance from superscalar?

3. [8] Suppose we have a *128-core processor*. For each one of the following scenarios, indicate the *maximum number of threads* that can be executed *at the same time* and explain, in 1-2 sentences, how you reached that number.

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

4. [2] We discussed Moore’s law in class. What is the importance of it?

5. [2] Why writing parallel code on multicore processor can help hiding memory latency?

6. [6] Can purely sequential programs benefit from multicore processors? If not, justify. If yes, give two scenarios showing how.

7. [2] We said in class that load imbalance, i.e. makes one core do way more work than others in a parallel program, is bad. Why is that?