Seminar 4
Parallel Processors and Programs

Computer Organization and Components / Datorteknik och komponenter (IS1500), 9 hp
Computer Hardware Engineering / Datorteknik, grundkurs (IS1200), 7.5 hp

KTH Royal Institute of Technology

Introduction

The purpose of seminars is to enable active learning of the more theoretical tasks that are typically part of the final written exam. Seminars are optional. However, we strongly recommend that you perform these seminar exercises and attend the seminar.

Rules. You may receive up to 1 extra point on the fundamental part of the written exam if:

- you make an honest *attempt* to solve *all* the seminar exercises on your own. You may discuss the exercises with your friends, but you are not allowed to copy any solutions from anyone or anywhere. You need to have written down a potential solution on all assignments. You are not allowed to skip some exercises, you need to try to provide a solution for all exercises.
- you write down your solutions *by hand* on this exercise form. You are not allowed to hand in machine printed solutions, copies, or handwritten solution on another paper format.
- you bring your solution *personally* to the seminar and attend the whole seminar. This means that you are not allowed to hand in a solution on behalf of someone else.
- you need to have signed this form before you hand it in.
- you are not allowed to attend the seminar if you are not bringing a solution, that is, if you do not bring this form filled out with your own solutions, you cannot attend the seminar.
- you must come to the seminar on time when it starts. If you are not there from the beginning, the assistants may refuse that you participate in the seminar.

Note that the extra point is only valid on the next ordinary exam, and the following two retake exams. During the seminar, the teaching assistant or teacher will go through the solutions and you will correct the solution done by another student. You need to have received at least 50% of the total number of points to pass the seminar. In such a case, you get one extra bonus point on the exam. We recommend that you take a photo of your solutions before you hand it in.

By signing the following, I hereby guarantee that I follow the rules above.

Your name (printed): _____

Signature: _____

Date: _____

Exercises

1. For the following four statements, state if the statement is correct or incorrect. If you think it is incorrect, explain why and what a correct statement would be.

   (a) Around 2006, *Moore's* law did not hold anymore. As a consequence, larger processor manufacturers started to produce multicore processors.

   (b) The term *power wall* typically means that the maximum clock speed has been reached and that chips would be too hot if they were clocked at higher frequencies.

   (c) Programming concurrent programs that achieve high parallel execution is a hard, but recently solved problem.

   (d) Before the multicore era, speedup by executing programs in parallel was basically not possible.

   Your solution:

2. Assume that you have a program that would take 20s to execute if you were running the program on infinitely many cores (if that was possible). Assume further that if you ran the program on 8 cores, you would get a speedup of 4. Assume also that N number of cores result in N times improvement for the part of the program that is possible to parallelize. What would then the speedup be if you executed the program on 20 cores?

   Your solution:

3. Consider the MIPS code below:

```
1    addi $t0,$t1,1
2    sub  $t1,$s0,$s1
3    add  $s1,$t0,$t1
4    xor  $t0,$t1,$s1
```

   (a) Assume that we execute the code on a superscalar MIPS-processor with four functional units. What is then the maximal *instructions per clock cycle (IPC)* for the code snippet? Note that the IPC is computed as the average instructions per clock cycles when executing the above four instructions.

   (b) What kind of dependency must an out-of-order processor take care of if it switches the order of the `addi` and the `sub` instructions? How can the processor resolve the hazard that occurs if the instructions are swapped?

   Your solution:

4. Categorize the following keywords and concepts into one of the following two scenarios. Choose the scenario that fits best.

   Keywords and concepts: SIMD, MIMD, hardware multithreading, VLIW, cache coherence, Advanced Vector Extension (AVX), MapReduce, and dynamic multiple issue.

   - Scenario #1: A computer with a shared memory superscalar multicore (8 cores) processor with simultanious multithreading (SMT). The processor has separated L1 caches and a common L2 cache. The computer uses a modern Linux operating system.
   - Scenario #2: A cluster system with 1000 computers connected by an Ethernet network. Each computer has an uniprocessor with static multiple issue and data-level parallelism at the instruction level. The processors have separated instruction and data caches. The cluster has software for massive big data computations.

   Your solution:

5. Explain the differences between a *process* and a *thread* for a modern operating system. The answer should include and clarify the following keywords: virtual memory, memory protection, files, global data, stack, and multicore.

   Your solution:

6. Assume that you are writing a multithreaded program in an imperative C-like language. You have one function `foo()` that takes zero arguments and does not return a value, but it has a side effect. This function is called from several threads. Explain and give an example with pseudo code for how you can make this function thread safe.

   Your solution:

Corrected by _____. Total number of points: _____