Computer architecture

Homework week 8-9

Instructions

Submit by e-mail to the lecturer, as a PDF document with your name and student ID near the beginning. You can work in groups of 2, however, use different groups than week 5, 6 and 7. Use the English language. Deadline: Nov 4th, 23:59.

> **Note**
> This is a 2-week assignment and will thus count double.

Question 1 (2pt)

Write in pseudo-code the behavior of a 2-way set-associative cache upon receiving a load request from below (processor-side). Assume write-through (no evictions needed).

A request is parameterized by an address (`addr`) and a size (`size`).

You can use in your pseudo-code:

- `S`, the base-2 logarithm of the number of sets (number of sets = 2^S)
- `L`, the base-2 logarithm of the size of a cache line in bytes (cache line size = 2^L bytes);
- `data[i][j]` the entire line contents for line `j` in set `i`;
- `data[i][j][k]` the byte at position `k` in `data[i][j]`
- `tag[i][j]` the tag for line `j` in set `i`
- any additional information you need indexed by set and/or line address, if you explain beforehand what it means and what it is used for.

Question 1 - Virtual Address Translation (2pt)

The Niagara T3 processor has 8 hardware threads per core: the pipeline can interleave instructions from 8 different program counters stored on the processor itself.

We consider a function in C which uses 1/8 of the pipeline capacity. In theory if we run this function on 8 hardware threads simultaneously on the same core we could use the pipeline close to 100% efficiency.

Indeed, experimentally[1] if we use 8 software threads in the same Unix process the performance approaches 100%. However in the same experiments we see that after 4

processes are created each with one software thread and efficiency approaches 50%, the performance per thread *decreases* as the number of processes grow.

Determine what hardware component may be causing the issue and explain why.

> **Note**
>
> To answer this question you must understand the difference between multiple software threads within one process and multiple processes. If in doubt, contact your lecturer and/or assistants.

Question 2 - Delay Slots (3pt)

1) Research and explain in your own words what *delay slots* are in relation to processor pipelines, and how they can be (or are) used.

   Give examples of processors/architectures that have *branch delay slots* and examples that have delay slots for other types of instructions.

2) The SPARC instruction set architecture (ISA) supports a so-called "annul bit" on branch instructions. Explain in your own words what is the purpose of this bit and give an example use.

Question 3 - Predication (3pt)

1) The ARM ISA up to version 7 is well-known for one of its flagship features, *instruction predication*.

   Explain in your own words:

   - what instruction predication is;
   - why it is useful, ie. what problems it intends to avoid;
   - where it is handled in the processor pipeline;
   - its limitations.

   Provide examples to illustrate as necessary.

2) In many graphical processing units (GPUs) a special type of processor, called a texture shader, provides an instruction set which allows programmers to encode many types of computations, not only graphical rendering. They are therefore often advertised as "general-purpose GPUs". Early versions of these processors did not support conditional branches; instead they supported only predication.

   Give an example computation that would be difficult or inefficient to run on such a processor.

[1] Little story: this is a real problem that was found during scientific research at the CSA group in Amsterdam. This finding was used by Oracle to improve the next generations of Niagara (T4/T5).