Comparative Architectures Supervision Problem (Set 2)

Wei Song 13/02/2016

The following problems are mostly open-ended. Diagrams or charts are not needed in the answers. Therefore it is highly recommended to type the answers. Please keep your answers organised and tidy. Most of the questions are taken from the supervision notes from the lecturer. If you are not sure about the answers, you could look up from the Hennessy and Patterson book or directly from the Internet. If you could not find the answer or would like a discussion during the supervision session, please label it in your answer.

Lecture 5 & 6. Superscalar

Read the paper about the ALPHA 21264 superscalar processor and answer the following question.

Link: https://www.cis.upenn.edu/~milom/cis501-Fall09/papers/Alpha21264.pdf

Q1. For the Alpha 21264 processor, two architectural techniques boosted the instruction fetch efficiency respectively, one is line and way prediction, and the other one is branch prediction.

1. The line and way predictor predicts the line-way of the instruction cache that will be accessed in the next cycle. Please read the paper and explain how it could hide the long delay of the branch predictor and why the line and way predictor could be trained by the results of the branch predictor?
2. What patterns in branches are better predicted by global branch predictors and what patterns are preferred by local branch predictors?

Q2. Register renaming exposes instruction-level parallelisms since it eliminates unnecessary dependencies. Out-of-order execution then exploits the benefits and deploys them on multiple execution units.

1. Which dependences are removed and how renaming could remove the hazards caused by these dependences?
2. Instructions could be issued out of order, why do we need to fetch and commit (retire) instructions in order? What problems occur when committing instructions out of order?
3. When a mispredicted branch or an exception is detected, how would the register map be reverted to the latest valid state? What additional hardware is needed?

Q3. The out-of-order execution of ALU instructions in a superscalar processor is only constrained by the availability of functional units and true data dependencies. Why must the out-of-order execution of memory instructions be constrained further? (2009 Paper 7, Q 7)

Lecture 7. VLIW

Q4. What components in typical superscalar processors are no longer needed in VLIW processors?

Q5. Why might dynamic binary translation be particularly useful in the case of a VLIW machine? (Sup-work Q 7.2)

Q6. Some VLIW processors contain additional hardware to permit memory reference speculation. What optimisations does memory reference speculation permit? Briefly describe the additional hardware required to support this type of speculation. (2008 paper 8, Q 5)

Q7. What advantages does dynamic scheduling offer when compared with static scheduling in a superscalar processor? (2010 paper 8, Q 5)

Lecture 8. Thread level parallelism

Q8. What are the advantages of exploiting Thread-Level Parallelism (TLP) in addition to Instruction-Level Parallelism (ILP)? (Sup-work Q 8.5)

Q9. Briefly describe the differences between coarse-grained, fine-grained and simultaneous multithreading. (Sup-work Q 8.2)

Q10. Describe techniques for reducing the thread switch penalty in a coarse-grained multithreaded processor. (Sup-work Q 8.4)