ELL 782/SIL 618

Computer Architecture
Major Exam, Sem I, 2021-22

Instructions

1. Submit a single pdf file. The order of answers should be the same as the order of the questions.
2. Scanned copies of handwritten submissions are not allowed. You need to type your answers; use software to draw your diagrams, and then create the pdf file. Preferably use LaTeX and inkscape. In inkscape, save your diagrams as pdf files, and include the pdf files in LaTeX. Avoid .bmp, .jpeg and .png files. No screenshots.
3. The pdf file must mention your name, entry number, and the course you have registered for (ELL 782 or SIL 618) on the first page. This is **mandatory**.
4. We will use Turnitin to check for copying (plagiarism). Turnitin is the world's best software in this category. It can find matches between answer sheets submitted by the students in the class, answer sheets submitted by ex-students, and all sources on the web. If we detect any instance of plagiarism, then the student gets zero marks in the minor and will be referred to a disciplinary committee.
5. All the questions are for 10 marks each.

Questions

1. Suppose we want to design an OOO processor with a wider pipeline and faster clock, then we need to scale all the relevant microarchitecture components accordingly.

   a. Find out whether brute-force approaches for scaling the LSQ (increasing the number of entries) will work or not. Provide a proper explanation.
   b. Propose an approach for scaling the LSQ efficiently both in terms of its capacity and search bandwidth.

2. We need to create a new instruction called MCAS (multi-word CAS). (Refer to the textbook Pg. 9.117 for the pseudocode of the multiple compare-and-set (MCAS instruction))

   a. Provide a hardware implementation of MCAS that makes it *appear* to execute atomically (instantaneously). What changes do we need to make to the ISA, the pipeline, and the memory system? Note that we have to introduce a simpleRISC instruction called MCAS. How do you give it so many arguments?
   b. Use MCAS to implement lock and unlock functions. Show the code and justify it.

3. In an OOO processor, value mispredictions result in costly pipeline flushes. List out the strategies to minimize the number of mispredictions without adversely affecting the training time and coverage of the value predictor. Explain in detail with diagrams.

4. Most secure architectures such as INTEL SGX have a threat model that assumes that the processor is trusted and the OS is untrusted. Such a Trusted Execution Environment ensures

that the applications run securely even in the presence of a malicious OS. However, it lacks trusted I/O paths, and thus I/O messages need to pass through the OS. It is possible for the OS to maliciously read and modify I/O data. Propose a solution to this problem. Prove why it works and is efficient.

5. What are inclusive caches? Is it necessary to make caches inclusive always? If we make the last-level cache exclusive, what changes do we need to make in this cache's insertion and replacement policy?