

COMPUTER SCIENCES DEPARTMENT
UNIVERSITY OF WISCONSIN – MADISON
PH.D. QUALIFYING EXAMINATION

Computer Architecture
Qualifying Examination

Fall 2015

GENERAL INSTRUCTIONS:

1. Answer each question in a separate book.
2. Indicate on the cover of *each* book the area of the exam, your code number, and the question answered in that book. On *one* of your books list the numbers of *all* the questions answered. *Do not write your name on any answer book.*
3. Return all answer books in the folder provided. Additional answer books are available if needed.

SPECIFIC INSTRUCTIONS:

Answer all of the following **SIX** questions. The questions are quite specific. If, however, some confusion should arise, be sure to state all your assumptions explicitly.

POLICY ON MISPRINTS AND AMBIGUITIES:

The Exam Committee tries to proofread the exam as carefully as possible. Nevertheless, the exam sometimes contains misprints and ambiguities. If you are convinced a problem has been stated incorrectly, mention this to the proctor. If necessary, the proctor can contact a representative of the area to resolve problems during the *first hour* of the exam. In any case, you should indicate your interpretation of the problem in your written answer. Your interpretation should be such that the problem is non-trivial.

1. Virtual Memory

Virtual address *synonyms* can complicate the design of the address translation mechanism and/or the cache hierarchy.

- a) Describe what is meant by a virtual address synonym. Give an example.
- b) Describe what problem(s) can arise from virtual address synonyms if the first level cache is not designed appropriately. Give an example.
- c) Discuss *hardware-only* solutions for preventing virtual address synonyms from being problems.
- d) Discuss solutions that require *software-visible* changes to prevent virtual address synonyms from being problems.

2. Deadlock

Consider implementing a directory coherence protocol in a multicore chip consisting of nodes (core, private L1 cache, and bank of a shared last-level cache (LCC)) interconnected with a two-dimensional grid. State any additional assumptions that you need to make.

- a) What are deadlock considerations for routing a message in the interconnect?
- b) How might you resolve the above considerations?
- c) Even with a deadlock-free interconnect, are there additional deadlock considerations for the coherence protocol? If so, how might you resolve them?

3. Critical Sections

Critical sections are a widely-used synchronization abstraction used to control concurrent access to read/write regions of shared data. While architects have proposed a plethora of synchronization mechanisms over the years – including barriers, locks, full/empty bits, and many others – locks have been the most widely-used mechanism for enforcing critical sections.

- a) Give one simple implementation of a lock using pseudo-assembly language instructions (i.e., that are representative of a typical ISA, but need not be exactly the same syntax) that is sufficient to enforce a critical section.
- b) Discuss some hardware techniques to improve the performance of critical sections implemented using locks.
- c) Transactions are an alternative synchronization abstraction that is similar to critical sections. Discuss how a hardware transactional memory system might help to improve the performance of lock-based critical sections.

4. Thread-Level Speculation

- a) What is the basic idea of thread-level speculation?
- b) In an implementation of thread-level speculation on multi-core or multiprocessor systems what is the responsibility of the programmer, compiler, and hardware?
- c) How are the three classical types of data-dependences - RAW, WAW, and WAR -- between threads in a TLS system handled?

5. Branch Prediction

Wide-issue processors remain an interesting and promising direction for microprocessor design and research. To handle wider issue, we need to fetch more instructions every cycle. Branch-prediction has been considered essential to improve the rate at which we can fetch instructions.

- a) Sketch the design of a basic two-level branch predictor for a single-issue machine that fetches one instruction every cycle.
- b) Explain the challenges in building a branch predictor for a wide-issue machine (say 8-wide OOO) by contrasting with the previous simple branch-predictor for a single-issue in-order processor.

6. Specialization

In recent years there have been many proposals for building specialized architectures or specialized engines where some additional entity is integrated on-chip with a general purpose processor.

- a) Explain the basic motivation for such specialized architectures or accelerators in comparison to conventional OOO processors.
- b) Some examples of specialization to exploit data-level parallelism is the usage of SIMD, vector or GPGPU processing. What other types of specialization can be employed to improve the performance or reduce the energy consumption of applications.