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

Computer Architecture
Qualifying Examination

Spring 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

Consider an architecture like x86-64 that supports a 64-bit virtual address space with aligned pages of 4KB, 2MB, and 1GB via hierarchical page tables.

(a) What is the largest four-way set-associative L1 cache that can be indexed quickly with page-offset address bits that do not require translation? Why?

(b) Discuss how to support all three page sizes with one or more *fully-associative* translation lookaside buffers (TLBs)?

(c) Discuss how to support all three page sizes with one or more *set-associative* translation lookaside buffers (TLBs)?

2. Using Analytic Models

Analytic models are an evaluation technique that complements detailed simulation and system prototyping.

(a) Discuss the *advantages* of analytic models vs. alternative evaluation methods.

(b) Discuss the *disadvantages* of analytic models vs. alternative evaluation methods.

(c) As computers differentiate from generic, general-purpose systems to range from the small (e.g., embedded) to the large (e.g., warehouse scale), do you expect that analytic models for be used more, less, or about the same? Justify your answer.

3. Prediction

High-performance out-of-order superscalar processors use a variety of mechanisms to predict certain values in order to improve performance.

Approaches that have been widely studied in the literature are Branch Direction Prediction, Branch Target Prediction, and (Data) Value Prediction.

(a) What is Branch Target Prediction (not Branch Direction Prediction) and how does it seek to improve performance?

(b) What is (Data) Value Prediction and how does it seek to improve performance?

(c) In what ways are Branch Target Prediction and (Data) Value Prediction similar? In what ways are they different?

(d) Almost all high-performance commercial processors use at least one form of Branch Target Predictor, while almost none use (Data) Value Predictors. What do you think are the reasons for this dichotomy?

4. Memory Consistency Models and Write buffers

A memory consistency model is an architectural specification of the permissible orders of loads and stores from different processors in a multiple processor system. A (non-speculative) write buffer is a structure that holds the value of a committed (aka retired) store before that value is placed in the cache.

(a) Total Store Order (TSO) is the memory model specified in the SPARC and x86 architectures. What orders are permitted by the TSO memory model?

(b) Explain why TSO allows the use of (non-speculative) write buffers (aka senior store buffers)? Describe what conditions must hold before a processor supporting TSO may place a store's value in a write buffer. Describe what conditions must hold before a processor supporting TSO may place a store's value in a coherent cache.

(c) Explain why TSO does NOT allow the use of (non-speculative) *coalescing* write buffers?

(d) Some weaker models do allow the use of coalescing write buffers. Give one example of such a model and explain why coalescing write buffers are legal in that model.

5. Data Parallel Architectures

Data parallel architectures seek performance and efficiency by applying the same operation to multiple data elements. Examples include classic array processors (e.g., Illiac IV and Connection Machine), classic vector processors (e.g., Cray I), modern short vectors (e.g., MMX/SSE/AVX), and general-purpose graphics processors (GPGPUs).

(a) Explain the ways that these *architectures* are fundamentally similar and fundamentally different.

(b) Explain the important ways that these *implementations* are similar and different.

(c) GPGPUs have recently become the most common data parallel architecture. Do you believe this has more to do with the differences in architecture, implementation, or something else?

6. Speculative Execution and Transactional Memory

Microprocessors have long employed speculative execution techniques to improve instruction-level parallelism. The earliest form of speculative execution was branch speculation, where an outcome of a branch was predicted and successive instructions executed in a speculative manner, with corrective action taken to recover from a mis-speculation. More recently, some microprocessor have taken speculation to a different level, in the form of transactional memory. Here a chunk of code (a transaction) is executed speculatively, in the expectation that its execution is atomic (occurs at once) with respect to the execution of other threads, with corrective action taken in case of a mis-speculation.

(a) Discuss how techniques for detecting a branch misspeculation are similar to, and different from, techiques for detecting a transactional conflict.

(b) Discuss how techniques for recovering from a branch misspeculation are similar to, and different from, techiques for recovering from a transactional conflict.