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

Computer Architecture
Qualifying Examination

Fall 2016

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. Short Pipeline Design

Assume that you are charged with design of a core pipeline that is energy-efficient, uses fewer transistors than aggressive pipelines, and—within these constraints—seeks the best performance possible. The option you will explore in the answer to this question is a pipeline that uses *three* pipeline stages. For parts (a) and (b), assume that multithreading is ***not*** supported. State any additional assumptions that you need to make.

a) What micro-architecture structures present in aggressive pipelines would you consider retaining, downsizing or eliminating for a three-stage pipeline? Why?

b) Discuss the factors that affect how you would partition the micro-architecture structures among three pipeline stages. Propose a specific partitioning and explain why you chose this one.

c) How would you change your non-multithreaded three-stage pipeline if you were asked to add multithreading support?

2. Cache Write Policies

The write policy is an important parameter in cache design. Early shared memory multiprocessors used write-through cache policies. Later they found writeback cache policies to be a better design option. When chip multiprocessors (CMPs) emerged, most also came with multi-level caches and they used write-through caching policies for their level-1 (L1) caches, but used writeback for their level-2 (L2) caches (i.e., stores update both the L1 and L2 caches, but not main memory).

a) Discuss the trade-offs between write-through and writeback caching policies for multiprocessors. What factors favor one over the other?

b) Explain why shared-memory multiprocessors evolved from using write-through to writeback caching policies.

c) With emerging technologies and future workloads, how do you see write policies being used in the different levels of future CMP caches?

3. Prefetching and Energy

Most modern multi-core processors support one or more forms of cache prefetching, using ISA extensions to allow software to direct which data to prefetch, hardware structures to detect and predict memory access patterns, or both. Prefetching has long been considered primarily a means to improve performance, but in today's power constrained world, prefetching has the potential to either help or hurt energy efficiency.

a) Discuss reasons that prefetching can help improve energy efficiency.

b) Discuss reasons that prefetching can hurt energy efficiency.

c) Discuss the trade-offs between how you would design a prefetcher to optimize performance versus to optimize energy efficiency.

4. Message Passing

Message passing enables parallel programming without a shared address space wherein data is communicated among processes with SEND and RECEIVE routines. However, indiscrimant use of messages can lead to poor performance.

a) For best performance, programmers are often advised to avoid small messages. Why?

b) What other programming techniques would you recommend for better message-passing performance? Why?

c) Message passing libraries, such as MPI, often include alternative SEND routines. MPI_SEND, for example, always sends the value of its buffer argument at the time is was invoked even if the sending thread subsequently writes the buffer. On the other hand, MPI_ISEND permits the sending thread's subsequent buffer modifications to affect the contents to be sent. What are the advantages and disadvantages of these two types of SEND?

5. Reliability

a) What are transient (soft) errors?

b) It is claimed that all microarchitectural structures are not equally susceptible to the effects of transient and permanent errors. Why? Discuss by comparing and contrasting the susceptibility of the branch predictor, register file, issue logic, and different levels of the cache hierarchy.

c) Discuss what techniques, if any, might be appropriate for mitigating the varied effects of transient errors in the branch predictor, register file, issue logic, and different levels of the cache hierarchy.

6. Importance of Processor Core Design

In recent years it has become possible to put tens (or even hundreds) of relatively simple (or "smaller") processor cores on a chip. However, it is also possible to put fewer "bigger" processor cores that are not only more powerful than simpler cores but also require more design effort and resources. Some have argued that little effort should be expended on the design of a processor core, and the effort should instead be concentrated on other components. Others have argued that the design of a processor core is still critical and deserving of significant design resources.

(a) What factors determine the number and types of cores, and other components that should be deployed in a given processor chip design?

(b) Do you believe that processor core designs will continue to be deserving of considerable design resources and effort in the next decade? Justify your answer.