

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

Computer Architecture
Qualifying Examination

Spring 2014

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 Caches

Most L1 caches today are *physical*: indexed and tagged with physical addresses.

- (a) How do most L1 caches today avoid the latency penalty of logically performing address translation before an L1 cache lookup? Under what constraints?
- (b) What are the pros and cons with switching to a virtual L1 cache? Be sure to consider single-core and multicore issues.
- (c) What are the pros and cons with switching from physical to virtual for (i) a private L2, (ii) an L2 shared among a few cores, or (iii) a last-level cache (LLC) among all cores on a chip?

2. Error detection and correction

In the 1990's, microprocessors went from having no memory error detection or correction, to using parity to detect single-bit memory errors, to using SECDED ECC codes to detect double and correct single bit memory errors. Similarly, over the past decade, microprocessors have gone through the same progression for L2 caches. Some systems are even using ECC codes for L1 caches and registers.

- (a) Discuss the tradeoffs with using no coding, parity codes only, and SECDED ECC codes at different levels of a *uniprocessor* cache hierarchy.
- (b) How does the choice of code and/or code word size interact with other cache hierarchy management policies, e.g., writethrough vs. writeback and inclusion vs. exclusion.
- (c) Conventional ECC codes are designed to tolerate errors that arise while values are stored in the memory hierarchy. However, errors may also arise while communicating values between levels of the memory hierarchy (or between the processor and L1 cache). Discuss how these two problems – tolerating memory and communication errors – are similar and how they are different.

3. On-Chip Interconnects

A multicore chip must interconnect components such as cores (with private caches) and shared cache banks. Early multicore designs used buses that broadcast requests to all destinations, while later and future chips consider crossbars, rings, grids, tori, etc.

As a multicore chip scales to more, smaller transistors:

- (a) How does the goal of best performance affect the choice of interconnect topology?
- (b) How do fault-tolerant considerations affect the choice of interconnect design?
- (c) What other considerations affect interconnect design? How?

4. Special Instruction Caches

Typical instruction caches simply store instructions to improve the latency of instruction fetching. However, other variants of instruction caches have been proposed to facilitate other aspects of instruction processing. For example, decoded instruction caches store instructions in some decoded form to improve instruction decoding. Other examples include storing additional information, recoding the instructions, and/or rearranging the information.

- (a) Consider an instruction set with variable length instructions. You want to design an instruction cache that adds additional information to each instruction or group of instructions. What types of information would you consider adding? Why? Discuss the tradeoffs.
- (b) Consider an instruction cache that recodes the instructions into a different format. Why might this be advantageous? What are the drawbacks of such a scheme? Discuss the tradeoffs.
- (c) For future processors, do you see the special instruction caches, like those in parts (a) and (b) above, becoming more or less common? Discuss your reasoning.

5. Memory-Level Parallelism

Achieving parallelism in the execution of instructions, especially long-latency instructions, is a key to achieving higher performance. Since arithmetic operations, including common floating-point operations, can be implemented with small latencies, memory operations are the dominant form of long-latency operations that occur in most common applications.

- a) Techniques to achieve the parallel execution of arithmetic instructions have some limitations when they are applied to achieving *memory-level parallelism*—the parallel (overlapped) execution of memory instructions. What are these limitations? Discuss ways of enhancing these instruction-level parallelism techniques to make them more amenable achieving at achieving memory level parallelism.
- b) What other techniques have architects proposed to improve memory level parallelism? Describe some techniques and discuss their pros and cons.

6. Architecture and Power

RISC/CISC instruction sets have been the dominant class of ISAs in the literature and commercially. In these ISAs, instruction-level parallelism or other forms of parallelism has been extracted by the underlying microarchitecture.

However, it has been argued over the years in academic work and in some commercial products that exposing parallelism in the ISA has many benefits and can provide power efficient execution, lower complexity processors, or higher performance.

- a) Explain one technique that has been proposed for exposing more information in the ISA and how it allowed obtain higher performance than RISC/CISC based machines.
- b) Argue that instruction set design and exposing more information in the ISA has a big role in the future in addressing the challenges of power efficiency.