SPRING 2011
COMPUTER SCIENCES DEPARTMENT
UNIVERSITY OF WISCONSIN—MADISON
PH.D. QUALIFYING EXAMINATION

Computer Architecture
Qualifying Examination
Monday, January 31, 2011

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. Combinational Logic Design of Multiplier

Implement an unsigned combinational multiplier whose inputs are two 8-bit binary numbers and which outputs a 16-bit binary number. You are provided with simple logic gates (NOT, AND, OR, XOR, etc.) and a 1-bit full adder. No gate can have more than four inputs. You will be graded on correct and hierarchical design.

a) Implement the above multiplier using standard binary multiplication (i.e., summing partial products).

b) What is the delay along the critical path? Assume all the simple logic gates have a fixed gate delay T (i.e., the delay is the same regardless of whether the gate has 1, 2, 3, or 4 inputs). Assume the delay of the full-adder is 3T.

c) Discuss optimizations for making this multiplier faster.

2. Caches and Power

Caches have long been an important part of a memory system. For many workloads, caches reduce average memory latency and reduce bandwidth demand, and thus contention, on lower levels of the memory hierarchy (e.g., an L1 cache reduces contention on an L2 cache, and a last-level cache reduces contention for memory).

a) Explain why caches tend to reduce average memory system energy (for a given computation).
b) Discuss whether caches only tend to reduce memory system energy or whether they also tend to reduce memory system power.
c) Caches have long been designed with a primary goal of improving performance. Discuss how cache designs might change as power and energy become more important design constraints than raw performance.

3. SIMD Processing

The Single Instruction Multiple Data (SIMD) processing paradigm is one that has been realized in a variety of different ways over the years. Examples include vector processors (e.g., Cray-1), multimedia instructions (e.g., Intel MMX and SSE), and graphics processing units (GPUs) (e.g., NVIDIA Tesla).

a) Explain how the SIMD paradigm is realized in the architecture of vector processors, multimedia instructions, and GPUs.
b) What are the disadvantages of the SIMD paradigm?
c) Despite these disadvantages, the SIMD paradigm has continued its presence in commercial hardware. Discuss why the paradigm will continue to be important and relevant for future architectures.

4. Technology Trends and Reliability

Recent research and the ITRS Roadmap highlights reliability and variability as important constraints for future microprocessors. One specific issue in reliability is the increase in the timing guard-band required to cope with increasing sources of variability. Specifically a timing-guard band means an additional safety factor added to the processor's nominal clock cycle time, making it longer than dictated by the nominal design. Typically, the nominal clock cycle time is computed based on the expected logic delays (under certain temperature and voltage conditions) and then a timing guard-band delay is added to set the final clock cycle time. Increasing the timing guard-band reduces the chance that variations in the actual logic delays may cause a timing-related error, but makes the processor execute at a lower frequency and hence reduces performance.

a) Explain some of the physical phenomenon which introduce variability necessitating this timing guard-band.

b) Circuit-level timing-speculation helps reduce or effectively remove this timing guard-band. Explain how circuit-level timing speculation works as developed using the Razor flip-flop and address how it allows the processor to operate with a smaller guard-band.

c) What are some of the disadvantages or design complications introduced by the Razor technique of timing-speculation?

5. Multicore and multiprocessor scalability

Large-scale distributed shared memory systems were designed to scale to hundreds and even thousands of processors, e.g., the SGI Origin could scale to 512 nodes. Modern multicore (sometimes called manycore) processors are beginning to integrate nearly as many cores on a single chip, e.g., Tilera already ships a chip with 100 cores.

a) Describe how the communication and caching mechanisms in distributed shared memory systems allowed them to scale so large.

b) Which of these mechanisms, if any, from large-scale distributed shared memory systems make sense to adapt to future multicore (manycore) processors?

6. Out-Of-Order Processor Design

You are the leader of a processor design team that has succesfully designed a 4-way out-of-order processor with an instruction window size of 40 instructions.

Your boss wants you to design a 4-way out-of-order processor with a *very large instruction window size* (a window size of 512 instructions). In preparation for your first meeting with your team, you need to prepare a list of which microarchitectural subsystems may need non-trivial modification and the key challenges that need to be solved for each subsystem. For each challenge, identify one or more potential solutions that you will want your team to explore further.

Present your list of microarchitectural subsystems and challenges and discuss which of these challenges are more or less important to solve in order to make your processor a successful product.