FALL 1997
COMPUTER SCIENCES DEPARTMENT
UNIVERSITY OF WISCONSIN—MADISON
PH.D. QUALIFYING EXAMINATION

Computer Architecture
Qualifying Examination
Monday, September 15, 1997
3:00 – 7:00 PM
3345 Engineering

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 nontrivial.

1. Combinatorial Shifter

Consider a combinatorial shifter whose inputs are `op<1:0>`, `amount<3:0>`, and `data_in<15:0>`, and whose output is `data_out<15:0>`. Assume `op<1:0>` is 0 for arithmetic shift right, 1 for logical shift right, 2 for shift left, and 3 is a don't care. With `op<1:0>==1`, for example:

```
data_out<15:0> = data_in<15:0> >> amount<3:0>
```

(a) Implement the above design using hierarchical design. The base parts are gates (NOT, AND, OR, XOR, etc.), decoders, encoders and multiplexors. Base parts should have no more than four inputs (except 4-to-1 multiplexors are allowed). You will be graded on correctness first, clarity of hierarchical design second, and speed third.

(b) What is the number gates delays in the critical path through your design, assuming decoders, encoders and multiplexors are three gate delays?

2. Floating Point Arithmetic

Before the recent adoption of the IEEE standard floating-point format, most computer manufacturers had their own floating-point number formats. For example, IBM had a 32-bit representation, with a base of 16 and the VAX had a format with a "hidden bit."

The three key parameters in a floating-point number representation are: (a) number of mantissa bits (and their representation), (b) number of exponent bits (and their representation), (c) the base of the number system.

(a) How does the choice of the base affect the quality of the floating-point number and the challenges in implementing it.

(b) Assuming a fixed number of mantissa + exponent bits, how does the tradeoff between mantissa and exponent bits affect the quality of the floating-point number and the challenges in implementing it.

3. Multiprocessors

For a new shared-memory multiprocessor design, sequential consistency must always be maintained. The processors will each have a single-level cache memory, with main memory distributed among the processing nodes. The processor will execute instructions out-of-order, with heavy branch prediction and other speculative techniques extensively employed. (Ignore writes to the instruction stream.)

(a) What are the challenges of implementing such a system with a simple bus that completes each transaction before beginning the next one?

(b) Can the system be implemented using a split-transaction bus?

(c) Should the processors use write-through or write-back? Defend your answer.

(d) Should the cache coherence algorithm use a write-update or write-invalidate protocol? Why?

4. Cache Design

The current trend toward ever faster processors and more modest increases in communications speeds requires that processors be able to tolerate ever longer delays between a request for data and its arrival. The choice of the unit of data transferred between memory and cache, i. e., the *block size*, can have a large impact on performance.

(a) What factors determine the optimal block size for uniprocessors with blocking caches?

(b) How are these factors affected by the introduction of a non-blocking cache?

(c) How are these factors affected in the multiprocessor environment? What additional factors are introduced?

5. Vector Execution

You are the execution unit architect of a vector machine, and are charged with designing an execution unit that would implement the following computation:

```
for i=1 to 64 do
S = S + X(i)
```

Assuming that: (i) `X(i)` is an array of 64-bit integers, (ii) transferring elements `X(i)` from wherever they are stored (for example a vector register) to the execution hardware is not a problem, (iii) overflow can be ignored:

(a) Propose two designs for the execution unit, one which achieves good performance with hardware comparable to a few adders (e.g., less than 5), and a second which optimizes performance without consideration of hardware costs. Discuss the cost and performance of each design

(b) Repeat part (a) for the following computation:

```
for i=1 to 64 do
X(i) = X(i-1) + Y(i)
```

Again, assume that: (i) `X(i)` and `Y(i)` are arrays of 64-bit integers, (ii) transferring elements `X(i)` and `Y(i)` from wherever they are stored (for example a vector register) to the execution hardware is not a problem, (iii) overflow can be ignored:

6. ISA Changes

Binary translators convert an executable from one instruction set architecture (ISA) to another. Assume that binary translation technology gets good, freeing architects to change ISAs.

(a) How would you change an instruction set to enhance performance, reduce cost, or both? Why? (There are no "correct" answers here. You be graded on how you justify your ideas.)

(b) What will binary translators need to do to make your ideas in (a) work? What is hard about making binary translators do what you need?