# SPRING 2002
# COMPUTER SCIENCES DEPARTMENT
# UNIVERSITY OF WISCONSIN—MADISON
# PH.D. QUALIFYING EXAMINATION

Computer Architecture
Qualifying Examination
Monday, February 4, 2002
3:00 – 7:00 PM
Room 2311 Chemistry

**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. Using Tri-State Buffers

Some logic families include tri-state (or three-state) buffers. These buffers typically have a data input DATAIN, a control input CONTROL, and a data output DATAOUT, such that:

- DATAOUT is 1 if CONTROL is 1 and DATAIN is 1,
- DATAOUT is 0 if CONTROL is 1 and DATAIN is 0, and
- DATAOUT is Z (high impedance) otherwise.

(a) Implement a 3-bit-wide four-to-one multiplexor using tri-state buffers effectively. Other basic gates (e.g., AND, OR, NOT) can be used as well. Assume the multiplexor has data inputs are IN0<2:0>, IN1<2:0>, IN2<2:0>, and IN3<2:0>, the control input is SELECT<1:0>, and the output is OUT<2:0>.

(b) Implement a 8-bit-wide bus with three input sources and two output destinations using tri-state buffers effectively. Other basic gates (e.g., AND, OR, NOT) can be used as well. Assume that BUS<7:0>'s data sources are SRC0<7:0>, SRC1<7:0>, and SRC3<7:0>, and control inputs are correspondingly ENABLE0, ENABLE1, and ENABLE2. Let BUS<7:0> and destinations DST0<7:0> and DST1<7:0> be driven to the value of SRCi<7:0> when ENABLEi is 1. DST0<7:0> and DST1<7:0> are undefined if no ENABLEi is asserted, and you can assume that at most one ENABLEi is asserted at any one time.

(c) Discuss implementing the logic of (a) and (b) without tri-state buffers. Discuss tradeoffs in using or not using tri-state buffers for multiplexors and buses.

## 2. Memory Latency

Differences in the rates of improvement of logic and memory speeds have resulted in ever increasing relative memory latencies. That is, the latency of memory operations, measured in processor clock cycles, has increased greatly over the years, and expectations are that it will continue to increase for the next decade. Architects have tried to address this problem with several techniques, for example, cache hierarchies, non-blocking caches, out-of-order execution, and prefetching.

(a) Discuss the various approaches that architects have used to deal with memory latency in the past, highlighting how and why they work, and highlighting the advantages and the limitations of each.

(b) Why are multiple approaches used simultaneously?

(c) What approaches do you expect architects to use in the future? Explain.

## 3. Data Parallel Programming Model

(a) What is the data parallel programming model? Be sure to include in your discussion how data are named (addressed), parallel work is coordinated (synchronization), and data are communicated (if necessary).

(b) Discuss implementing the data parallel programming model on a multiple-instruction multiple-data (MIMD) architecture of your choice. Please state your assumptions.

## 4. Memory Systems

One of the emerging distinctions among mainstream (i.e., Intel) processors is the application domain: desktop vs. server. Desktop processors generally run a single, or small number of threads at a time, though at least one of the applications may be highly demanding. Servers often run many processes at a time, and are usually more focused on throughput than on the performance of a single task. Desktop systems are much more sensitive to cost than server systems. Servers often require significantly larger amounts of memory.

Given these distinctions, how would you expect desktop and server systems to diverge with respect to:

(1) memory system parameters?

(2) multithreading capabilities?

(3) multiprocessing capabilities?

## 5. Power and Energy in Embedded Processors

Embedded processors are widely used today in portable devices such as cell phones and personal digital assistants (PDAs). Power dissipation and energy consumption are two increasingly important metrics for modern embedded processors. Power dissipation is typically measured in watts (i.e., joules per second) while energy consumption is normally measured in joules per unit work (e.g., joules per instruction).

(a) In some studies, researchers claim that power dissipation is a more relevant metric than energy consumption. In others, researchers claim that energy consumption is more relevant. Discuss when each metric is the more appropriate to use.

(b) Embedded processors use a number of different techniques to improve one or both of these metrics, including lowering the supply voltage, decreasing the clock frequency, and dynamically disabling parts of the chip. Discuss how these techniques help improve one or both of these metrics.

(c) Which of the techniques of part (b) are applicable to high-performance microprocessors in addition to embedded applications?

## 6. Instruction-Level Parallelism (ILP)

Over the past three decades, numerous studies have explored the limits to available instruction-level parallelism. At one time, researchers believed that computers were limited to issuing one instruction per cycle (i.e., the so-called Flynn Limit). Around 1990, Wall published two studies that asserted that the potential for instruction-level parallelism was quite limited. More recently, Wall and others have published additional studies that show that much more instruction-level parallelism may be possible.

The results of these studies differ largely because they make different underlying assumptions about what is possible to accomplish in hardware. Discuss the fundamental factors that impede instruction-level parallelism. Discuss the mechanisms that have been developed and refined to enable greater exploitation of ILP.