Department of Computer Science

Computer Architecture Qualifying Exam

Spring, 2001

This examination is open book and open notes. The questions are equally weighted. On all questions, the reasoning you use to obtain your answer is more important than the actual answer; if your answer to any question is simply a number without justification, it is wrong (even if the number happens to be correct).

1. Two cache organizations are being considered for a computer system. The computer system will use a 32 bit virtual address. Both of the caches under consideration will be able to respond to a memory request in one cycle. Happily, the two caches cost exactly the same amount.

   Cache A is a 4MB cache, with an eight byte line size, and is 4–way set–associative. The expected hit rate with this cache is 99%.

   Cache B is an 8MB cache, with a 64 byte line size, and is 8–way set–associative. The expected hit rate with this cache is 99.5%.

   The memory subsystem will have an eight byte wide bus regardless of the cache used. It will be capable of transferring bursts of data; the first eight–byte transfer in a burst will take 10 cycles, while each succeeding eight–byte transfer will take 1 cycle (this is the miss *penalty* – it is incurred in addition to the one cycle mentioned above, on a miss).

   A. Of the information I have given you regarding the cache and memory, what part is relevant, and what part is not relevant, in solving Part B?

   B. Which cache should be used?

2. Consider the following sequence of C code:

```
a = b * 2;
c = a * 4;
b = b - d;
if (a < b)
    d = c + 5;
else
    d - b - 4;
```

   Generate optimal code (in the sense of executing as quickly as possible) for an EPIC–style computer, assuming:

   - Every variable is in a register (not memory).
   - Every register is initially valid.
   - Every instruction takes the same amount of time in the pipelines.
   - You are able to execute an arbitrary number of instructions simultaneously. All

instructions denoted as starting simultaneously read their registers, perform their operations, and then write their registers, including predicate registers.

- You are able to use predicated instructions to replace `if-then-else` statements.

You may use any notation you want for the code you generate, so long as it is clear.

3. Consider the parallel execution of programs on four processors. The programs executed on the processors are as follows:

Processor 1	***Processor 2***	***Processor 3***	***Processor 4***
`if (i < j)` `    k = 3;`	`i = 2;`	`j = 2;`	`if (j < i)` `    l = 4;`

Assume that all variables have the value `0` before the program is executed, and that all are shared among all four processors.

A. Show that using the Strict consistency model, it is possible for `k` to be assigned the value `3`, and it is possible for `l` to be assigned the value `4`, but it is not possible for both `k` to be assigned `3` and `l` to be assigned `4` on a single run.

B. Show that using a Weak Ordering consistency model, it is possible for both `k` to be assigned `3`, and `l` to be assigned `4`, on a single run.

C. Show how to use a mutual exclusion operator of your choice to guarantee that `k` will be assigned `3`, but `l` will not be assigned `4`, under both consistency models.