

Name _____

Multiprocessors, Etc.
EE 7700-2
Final Examination
11 December 2003, 15:00–17:00 CST

Problem 1 _____ (35 pts)

Problem 2 _____ (15 pts)

Problem 3 _____ (50 pts)

Alias _____

Exam Total _____ (100 pts)

Good Luck!

Problem 1: (35 pts) In the code fragment below the loop is executed for many iterations. The address used by the first load is never the same as the address used by the second load. Line labels are provided for convenience.

LOOP:

```
A: lw r1, 0(r2)      # r1 = Mem[r2]  Loaded addresses unpredictable.
B: lw r3, 0(r1)      # r3 = Mem[r1]
C: addi r3, r3, 1    # r3 = r3 + 1
D: sw 0(r1), r3      # Mem[r1] = r3
E: bneq r2, r4, LOOP # if r2 != r4 then goto LOOP after next insn.
F: addi r2, r2, 4    # r2 = r2 + 4 (Part of the loop body.)
```

(a) Draw a dataflow graph covering two iterations. Use ellipses (\dots) to show how the dataflow graph would continue for additional iterations.

(b) Using the dataflow graph determine the ILP (ideal IPC) for a large number of iterations.

Problem 1, continued:

(c) Describe the potential for each of the techniques below to improve the performance of the loop over the base system described below.

The base system is 8-way superscalar and dynamically scheduled with a 128-entry reorder buffer and good branch prediction. There is a cache but performance is affected by cache misses. A load suffering an L2 cache miss must wait at least 100 cycles for the data.

If there is potential for improvement using the technique, **explain how that is achieved.** (Explain what is going faster, not details of the mechanism.) If there is little or no potential for performance improvement, **briefly explain why not.**

A system with a 1000-entry reorder buffer and other resources proportionately increased.

Prefetch. (Indicate the simplest kind that would be effective, if any.)

Pre-Execution

Problem 1, continued:

Implicit Multithreading (IMT)

Data Prediction

Dynamic Optimization (rePlay or optimization cache)

Problem 1, continued:

SMT (Simultaneous Multithreading), code can be modified.

Multiprocessor, code can be modified.

Problem 2: (15 pts) The code below runs on a multiprocessor using a write-invalidate, directory-based cache coherence protocol.

It takes more than 1 ns to send a message between any two processors, and any particular access can be completed in much less than an hour. The time needed to send a message from any processor to the home memory of address 0x1000 is identical. Initially the caches are empty.

- In time order, show the cache states, directory states, and messages that will be sent to execute the code below.

```
Proc 1, t = 0  
lw 0x1000
```

```
Proc 2, t = 1 hour  
lw 0x1000
```

```
Proc 3, t = 2 hours  
sw 0x1000 <- STORE, NOT LOAD
```

```
Proc 1, t = 2 hours + 1 ns  
lw 0x1000
```

```
Proc 4, t = 2 hours + 2 ns  
lw 0x1000
```

Problem 3: Answer each question below.

(a) (7 pts) Explain how a stride prefetcher determine prefetch addresses. If a table is involved (*Hint: there is*) describe how the table is indexed, what is stored in each entry, and how it is used to predict addresses.

(b) (7 pts) Show how the multiple branch predictor (MBP) covered in class can predict three branches in one cycle without having to do three consecutive PHT lookups.

(c) (7 pts) How might hierarchical sequencing of trace cache entries reduce the number of entries used in the trace cache? Show an example.

(d) (7 pts) Why might it be harder to get speedup with computation re-use than with data prediction?

(e) (7 pts) With confidence estimation data prediction accuracy can exceed 90% and result, by some measures, in good performance improvement. Then why is it not practical?

(f) (8 pts) Pre-execution and dynamic optimization using rePlay have similarities.

Describe a situation in which both pre-execution and dynamic optimization using rePlay could effectively pre-execute a load. Describe the steps involved.

Describe a situation in which pre-execution could effectively (though not perfectly) pre-execute a load but rePlay could not. Clearly indicate what rePlay's weakness is.

(g) (7 pts) You are developing a branch predictor that will be particularly accurate for program X . Assume that Shade, SimpleScalar, and SimOS could all simulate the same ISA.

Describe an advantage in using Shade over SimpleScalar.

Describe an advantage in using SimpleScalar over Shade.

Describe an advantage for SimOS.

Would an instrumentation system be appropriate?