CS252 – Graduate Computer Architecture

University of California
Dept. of Electrical Engineering and Computer Sciences
David E. Culler **Spring 2005**

Last name: _____ First name_____

I certify that my answers to this exam are my own work. I am allowed to use class materials and scholarly materials, but I have not discussed the questions with other members of the class, nor have been given guidance in answering the questions from others.

Signature: _____

The exam contains three questions intended to test (and improve) your understanding of computer architecture concepts. It is due at noon wed April 27. Submit it by email to culler@cs.berkeley.edu with the subject heading CS252 Midterm2.

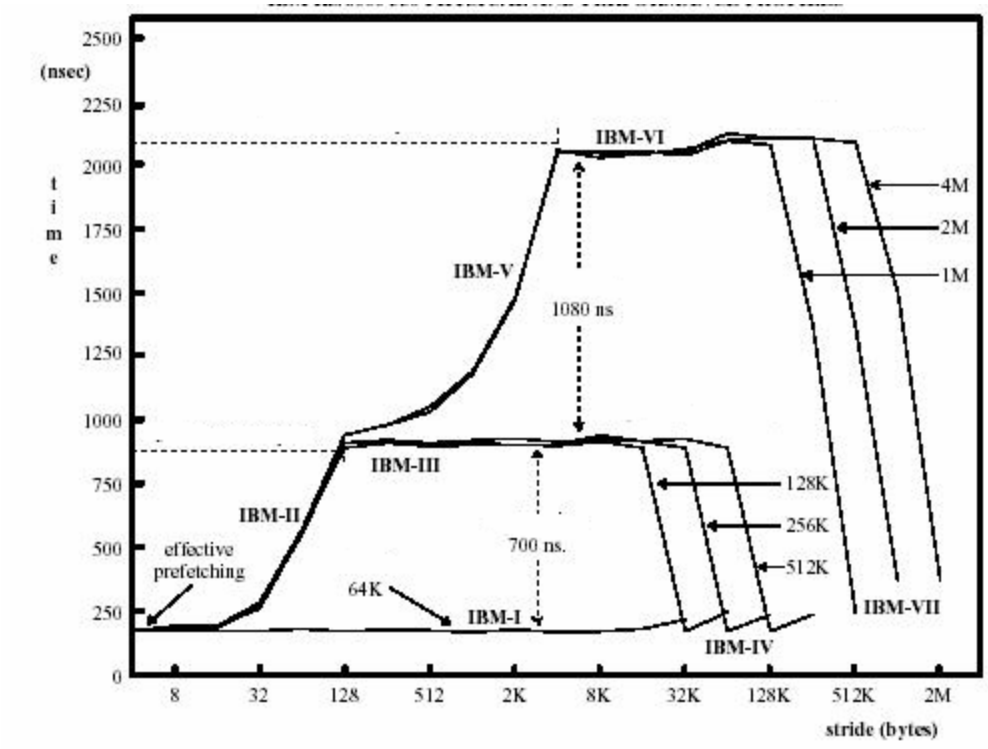
You are welcome to use course materials in answering the questions. You should find that sufficient. You must do your own work. If you utilize other materials, you must cite them and explain what role they play in your answer.

Problem	Points	Score
1	100	
2	100	
3	100	

Problem 1. In class we looked at average memory access time in terms of the statistical behavior of various workload on particular cache organizations. Here we are going to concentrate on average memory access time, but we will construct a program that generates a particular address stream in order to determine experimentally what is the cache organization of the underlying machine. We do this by timing a loop that steps through an array of size R with a stride of S performing a read-modify-write on each element. The loop is executed many times so that an average time per read-modify-write is obtained. Below we graph the average access time as a function of the stride (S) for several different array sizes (R) on some particular machine. From this simple measurement you can determine many characteristics of the memory system. When R is smaller than the cache size, all memory accesses hit. However, once R exceeds the size of the cache, some of the access will cause a miss. In some cases all of the accesses miss. Notice that we are not talking about cache behavior for programs in general, we are constructing a very special address pattern:

[A0=0, A1=S-1, .. Ai = iS mod R, …] .

If you'd like an analogy, we are subjecting the cache to a particular sawtooth signal and observing its response. Think about the access pattern and what hits and misses occur. Use the figure to determine all of the memory system characteristics listed in the table below. Explain your reasoning in each case.



Cache Size:	
Line Size:	
Cache miss penalty:	
Cache associativity:	
Page Size:	
TLB Miss Penalty:	
TLB Associativity:	

Problem 2. Vector and Parallel Processing

Consider the pseudocode loop nest below which traverses a two dimensional array. Assume all data elements are (A, B, T) are double precision floating point values. Assume that the arrays are stored row-major order, so consecutive elements of a row are stored consecutively in memory.

```
for i = 1 to n do
   T = 0
   for j = 1 to k do
       T = T + A[i,j] + B[i]
       A[i.j] = T
   endfor
endfor
```

2.a. Transform this loop so that it will execute efficiently on the VMIPS vector machines described in Appendix and rewrite it in pseudocode so that it maps to the instructions in Figure G.3.

2.b. Translate the pseudocode to assembly language for VMIPS.

2.c. Assuming instead that you have a chip multiprocessor with several MIPS scalar processors, each with private first level caches and sharing a second level cache and memory system, give the pseudocode for how you would implement this loop nest. You may assume that you have a synchronization library with locks, barriers, fetch&add and the like. Assume that all eight threads have previously been created (forked), all are executing the same program, and all have a private stack. There is a shared global value PROCS which is the number of threads. Each is assumed to be running on a distinct processor. Each thread has a thread local variable MYPROC set to a unique value between 0 and P-1.

2.d. Contrast the vector and multiprocessor implementations.

2.e. Describe in general terms how this might be implemented using message passing where a portion of the array is allocated in each of PROCS local address spaces.

Problem 3. Write a 2-3 page essay contrasting the design, analysis, and hardware/software tradeoffs presented in "Evaluating the Imagine Stream Architecture" by Ahn et al (http://cva.stanford.edu/~gajh/publications/isca2004.pdf) with "Building a Robust Software-based Router using Network Processors" by Spalink et al. (http://www.eecs.berkeley.edu/~culler/courses/cs252-s05/papers/p216-spalink.pdf)