

CIS 775: Computer Architecture  
Fall 2003  
Combined Lab #3 (100pts) and Hw #4 (100pts)

Instructor: Parthasarathy

Due date: Friday 5 Dec. by 5PM to be handed over to the instructor or slipped under the door to his office [It needs to be in the office not in the mailbox].

1. The objectives of this lab are to study the effect of various parameters that affect the performance of memory hierarchy. This lab will require the use of the *sim-cache* simulator environment on the *matrix.c* program. Note you will need to re-copy the *matrix.c* program from `/usr/class/cis775/`.

In the following experiments, we will consider the miss rate for different cache organizations. Consider a multi-level memory hierarchy consisting of three levels: **direct-mapped** split L1 (data and instruction cache separate), L2 (unified L2) and main memory.

Let the total amount of space available for L1 caches (instruction and data) be 24 Kilobytes. You can have a split of 8KB:16KB, or 16KB:8KB for the L1 instruction and data caches. L1 hit latencies are 1 clock cycle.

The total amount of space for L2 is 256KB. L2 can be 2-way, 4-way or 8-way associative. L2 hit latencies for 2-way associative memories are 6 seconds. Due to the complexity of multi-way associativeness, there is a 2% increase in L2 hit time when going from 2-way to 4-way and 4% increase when going from 2-way to 8-way associativeness. The replacement policy may be random or LRU. Assume that the latency to memory is 20 cycles.

Now solve the following. Identify any assumptions you make and clearly underline them. Note that in some cases the solution may involve pen and paper calculations in addition to simulation results. Include all supporting information (from simulator output). Use graphs where appropriate.

- (a) Determine the optimal L1 split configuration. [15 points from lab]
- (b) Then determine the optimal L1 block size (vary blocks size from 16,32 and 64 bytes). Note the block size you choose for the I-cache may be different from the block size for the D-cache. Use only miss rates to determine your answer and ignore the effects on replacement time (use graphs to illustrate). [15 points from lab]
- (c) Using the best L1 set up from the above determine the optimal block size and associativity for L2 (remember L2 block sizes have to be greater than or equal to L1 block sizes). Vary the block size up to 128 bytes. Use graphs to illustrate where appropriate. You will need pen and paper for part of your solution. [35 points from lab, 25 points from homework]

- (d) Using the best L1 set up and for the the optimal block size, for different associativities determine the savings in number of misses if any when going from random to FIFO or to LRU replacement. Assume that the cost of LRU and FIFO increases the miss penalty of L2 by 5%, is this worth it. You will need pen and paper for part of your solution. [35 points from lab, 25 points from homework].
2. (20 points from homework) Consider a memory system with a capacity of  $2^{32}$  bytes. The processor system has a 2048-word cache with 8-word (each word being 4 bytes) block size.
- For the following four cache organizations, determine how many bits are used for tag, index, and block offset, respectively.
- (a) direct-mapped
  - (b) 2-way set-associative
  - (c) 8-way set-associative
  - (d) fully set-associativity
3. (30 points from homework) Consider the execution of the following loop. Each element of array A and B are word-oriented. Assume array A is stored starting with byte address 1000 (in hexadecimal) and array B with byte address 2000 (in hexadecimal).

```
for i:=0 to 127 do
    A[i] = A[i]+B[i]
```

- (a) (10 points) How many memory read and write references are used for *data* access in executing the above loop?
- (b) (20 points) Consider a 64-word direct-mapped, write-through, and no write-allocate *data* cache with 8-word block size. Assuming the compiler generates the assembly code for the above program in such a manner that A[i] is loaded first followed by B[i]. For this cache organizations, determine the number of read hit, read miss, write hit, and write miss memory references (accesses). Explain your answers.