

Introduction to Computer Engineering (E85)

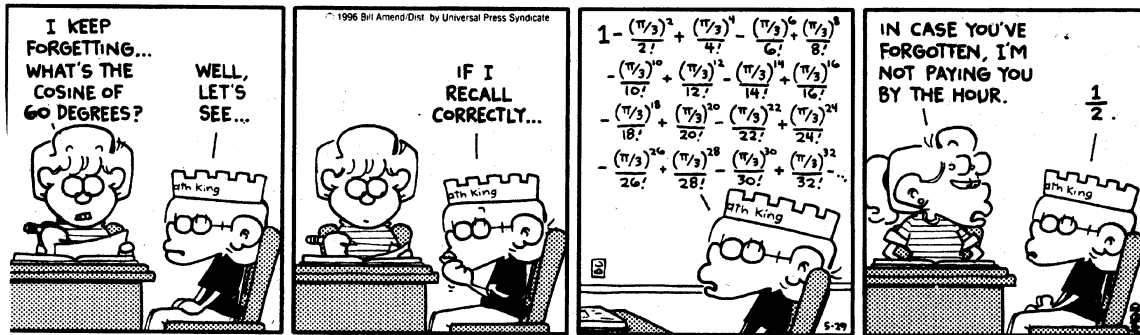
Harris

Spring 2001

Problem Set 10

Due: Friday, April 20

FOXTROT • Bill Amend



1) Cache Finger Exercises

- a) Come up with a sequence of addresses for a MIPS processor for which a direct-mapped cache of size 32 words, block size 8 words, outperforms a fully-associative cache with the same block size, using LRU replacement.

Suppose the block size is B 32-bit words, S is the number of sets, D is the degree of associativity (also called the set size or the number of ways), and addresses are made up of A bits. Refer to the text for these definitions. Assume that the cache is word addressed, i.e., the low two bits of the address are always 0.

- b) In terms of the parameters described above, what is the cache size in bytes? This number should include just the data portion of the cache.
- c) In terms of the parameters above, what is the total number of bytes required to store the tags?
- d) What are S and D for a fully-associative cache of size x bytes, with block size B?
- e) What is S for a direct-mapped cache of size x bytes, with block size B?

2) Cache Organization and Miss Rate

Consider the following repeating sequence of 1w addresses (in hex):

repeat 40 44 48 4C 70 74 78 7C 80 84 88 8C 90 94 98 9C 0 4 8 C 10 14 18 1C 20

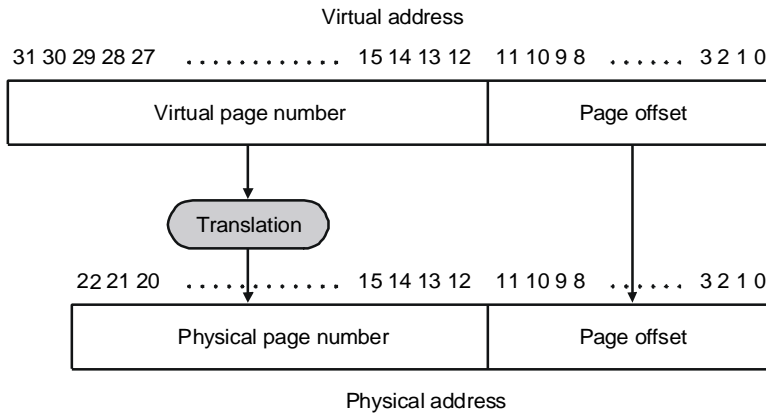
Assuming LRU replacement, determine the effective miss rate if the sequence is input to the following caches, ignoring startup effects (i.e. compulsory misses).

- ? Direct-mapped cache (D=1), S= 16, B = 1 word

- ? Fully-associative cache (D=16), B=1 word
- ? Two-way, set-associative cache (D=2), S=8, B=1 word
- ? Direct-mapped cache (D=1), S=8, B=2 words

3) Virtual Memory

Suppose we wish to design a virtual memory system. We would like to be able to address a total of 2^{32} bytes. We've got as much disk space as we want, but are limited to only 8 MB of semiconductor memory, i.e. to physical addresses that are 23 bits long. Assume in the following that we have decided to divide blocks of memory (both virtual and physical) into 4 KB pages.



- a) How many virtual pages are there in our system?
- b) How many physical pages are there in our system?
- c) Think of physical memory as a cache for virtual memory; virtual pages are “pulled in” to physical memory at different locations. We must come up with a way of mapping each virtual page into a physical page. If a “direct-mapped” scheme is used, i.e., if only the low order bits of the virtual page number are used to determine the physical page number, how many virtual pages are mapped to each physical page? Why is this “direct mapping” a bad plan? (HINT: try to provide an example of thrashing.)
- d) Clearly, a more flexible and dynamic scheme for translating virtual addresses into physical addresses is required. Suppose we use a page table to store mappings (translations from virtual page number to physical page number). How many page table entries will the page table for our system contain?
- e) Assume that in addition to the physical page number, each page table entry also contains some status information in the form of an R bit (set if this particular page table entry refers to a virtual page that is actually resident in physical memory), and an M bit (which is set if a particular physical page has been modified). How many bytes long is each page table entry? Sketch the layout of the page table. What is the total size of the page table in bytes? (Round up to an integer number of bytes).

4) Time

Please indicate how many hours you spent on this problem set. This will not affect your grade, but will be helpful for calibrating the workload for next semester’s class.