

# 15-740/18-740 Computer Architecture

## Homework 5

Due Friday, November 11, at 12:00 PM

### 1. Cache Organization

A very small, in-order embedded processor has a five-bit physical address space and a physically-addressed cache. Memory is byte addressable and the cache uses perfect LRU replacement. Assume the cache initially contains no valid data.

Suppose a program generates the following access stream and you observe the hit/miss outcome of the cache for each access as follows:

Address	Outcome
5	Miss
2	Miss
0	Miss
4	Hit
25	Miss
9	Miss
1	Hit
25	Hit
16	Miss
9	Miss
1	Miss

*Hint: For the following questions, keep in mind your answers need not be powers of two.*

- (a) What is the size of a cache block in bytes?

5M → 2M → 0M → 4H: two bytes.

- (b) What is the associativity of the cache?

Notice 9M → 1H → 25H → 16M → 9M: if cache were four-way set associative, the second access to 9 would have been a hit. Notice 25M → 9M → 1H → 25H: if cache were two-way set associative, the second access to 25 would have been a miss. Three-way set associative.

- (c) How much storage (in bytes) is required for the data portion of the cache?

Two bytes × three ways × four sets = 24 bytes.

- (d) What is the minimal amount of storage (in bytes) required for the tag portion of the cache (hint: this includes the address tag, modified data, valid data, and age information)?

Less efficient LRU: (Two tag bits + one dirty bit + one valid bit + two age bits)  $\times$  12 = 72 bits = 9 bytes. As discussed on the board, however, the LRU data for the blocks in a set can be represented using a single number which represents which of the possible LRU permutations pertains to the set. For our three-way set, there are six LRU permutations:

- i. (Most recently used, Next-most recently used, Least recently used)
- ii. (Most recently used, Least recently used, Next-most recently used)
- iii. (Next-most recently used, Most recently used, Least recently used)
- iv. (Next-most recently used, Least recently used, Most recently used)
- v. (Least recently used, Most recently used, Next-most recently used)
- vi. (Least recently used, Next-most recently used, Most recently used)

The number of bits required to store which of the six permutations a set follows is  $\lceil \log_2(6) \rceil = 3$ . So, for LRU with the permutation optimization: (Two tag bits + one dirty bit + one valid bit)  $\times$  12 + (three LRU bits \* four sets) = 60 bits = 7.5 bytes.

## 2. Cache Hierarchy

*For this question, assume there are two systems, A and B, each with two levels of on-chip cache. The systems are identical in every way except that system A employs an inclusive caching policy for its on-chip caches while system B uses an exclusive caching policy.*

- (a) You run a program on system A and find that it runs more slowly than when run on system B. Why might this be the case?

One possible reason is that the program has a large working set and makes inefficient use of cache space with multiple copies of the same data element at different levels of the hierarchy.

- (b) You run a program on system B and find that it runs more slowly than when run on system A. Why might this be the case?

One possible reason is that the program could generate a large number of coherency requests which could have been avoided by using an inclusive caching policy.