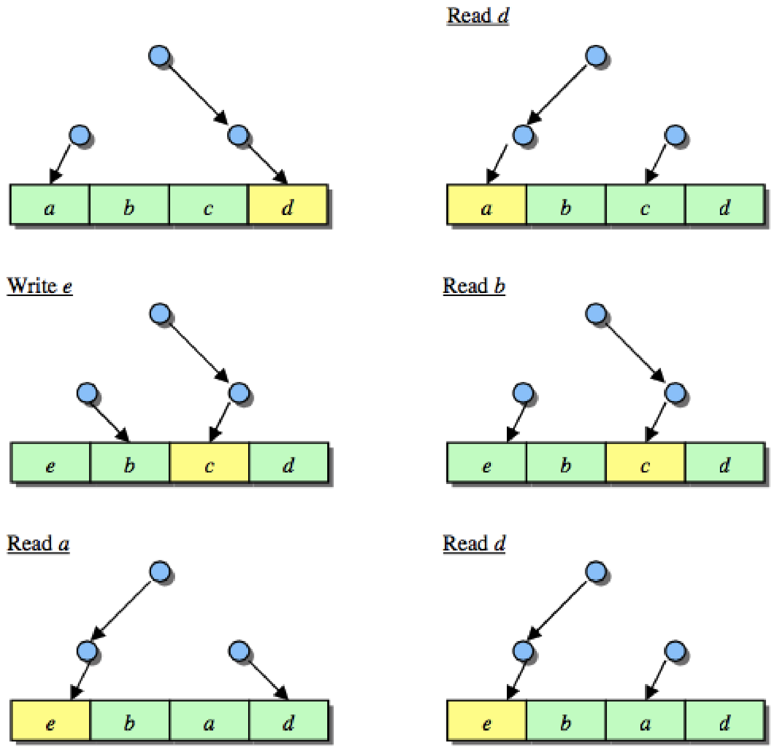
CS4100: Computer System Design
Lab Assignment #2
Department of Computer Science and Engineering
Indian Institute of Technology Madras

Submission deadline: 6th Sep, 2015

Designing a cache simulator: The objective of the project is to design a cache simulator that simulates the behavior of a real cache. Your design should be so generic that it can work with any cache line size, cache size, and associativity, provided as command line arguments. Memory traces provided by the Pin tool (refer to https://software.intel.com/sites/landingpage/pintool/docs/65163/Pin/html/) are used as the input to the cache simulator.



Read *d*

Write *e*

Read *b*

Read *a*

Read *d*

a. Implement tree-based pseudo-LRU replacement policy. Maintain a tree for each set in the cache, where leaf nodes contain cache blocks of a set and internal nodes store either 0 or 1. Value 0 indicates that we need to access left sub-tree and value 1 indicates that we need to access right sub-tree. Given a request, we search all the leaf nodes (irrespective of the internal node values) to determine whether the request is a hit or a miss. If it is a cache miss, to select a victim block, we follow the path from the root node all the way up to a leaf node based on the internal node values. Once the leaf node is selected, we place the requested block in the leaf node and flip the internal node values along the path from the leaf node to the root node. If the search operation results in a cache hit in a particular leaf node, we change the internal node values along the path from the leaf node to the root in such a way that none of the internal nodes points to a sub-tree that contain the leaf node. The following figures illustrate how the internal node values change on a cache miss/access in a 4-node tree. The following figures show the change in the internal node values after a cache access is made.

b. Write a program to multiply two integer matrices of size *n* x *n* (for a large value of *n*) using block multiplication method with a blocking factor of *s.* When *s* = 1, the entire matrix is treated as a block, whereas when *s* = *p,* the matrix is divided into p^2 blocks with each block of size (*n*/*p* x *n*/*p).* Note that the matrix elements are stored in row-major order.

c. Compare the conflict miss rate of the block multiplication method with the naive matrix multiplication method.