Cache Simulation Project

Specification

For this project, you will write a data cache simulator. This simulator should be parameterized in that it should support:

a. Various associativity from direct-mapped to fully associative
b. With variable cache size from 1KB up to 256KB
c. With variable cache line size from 16B up to 256B
d. For associative caches, use either FIFO or LRU replacement algorithm to select a victim

The parameter setting is like the following:

- -t trace_file
- -s cache_size (is the power of 2)
- -l line_size (is the power of 2)
- -a associativity (is the power of 2)
- -r FIFO or LRU

The cache size (-s) and line size (-l) are in bytes. For associativity (-a), 4 means four-way, 2 means two-way and 1 means direct-mapped. The replacement algorithm (-r) is either 0 for FIFO or 1 for LRU.

Trace file format

The file contains a trace of memory accesses executed from some benchmark program. Each line represents a 32-bit byte address by a memory access. It is in the following format:
0x00123458
0x00123450
0x00123460

Note: The cache write policy is write through plus write allocate, so that reads and writes are not distinct with respect to cache status changes.

Input command line

cache-sim –t trace1 –s 1024 –l 32 –a 2 –r 0

trace1

0xbfa437cc
0xbfa437c8
0xbfa437c4
0xbfa437c0
0xbfa437bc
0xbfa437b8
0xbfa437b8
0xbfa43794
0xb8088ea8
0xb8088eac

Output of your cache simulator

Trace file: trace1
Cache size: 1024
Line size: 32
Associativity: 2
Replacement: FIFO
Cache hits: 6
Cache misses: 4
Miss rate: 40%

Requirements

1. Bring your source code with comprehensive comments, that’s all.
2. Due date/Demo date: 2010/01/22 Fri.