DT021/4 Formal Assignment Semester 1 2017

Simulating a 2-Way Set Associative Cache

Problem:

Write a computer program in C or C++ that simulates the functioning of a 2-way set associative cache during a series of read operations. Assume a CPU with an 8-bit data bus and a 16-bit address bus. Assume instructions are 8 bits wide. The cache will initially be empty. A test file containing addresses is used as a test source. As each address is read from the file it is checked against the cache entries. A cache miss results in a cache line fill. Assume a cache line is 4 bytes. A cache hit should result in outputting of the relevant data entry. Hence the program should output the resulting address/data values and the effect of the operation on the cache. Use a second file to hold data values simulating DRAM.

Input:

Initially random data should be written into an array representing main memory. Your input test data should come from a file, consisting of a sequence of addresses. This simulates the behaviour of a CPU as it fetches program instructions from memory. The address can be represented by a 16-bit integer, split into an upper tag of 6 bits, 6-bit set no. and a 2-bit word select as follows:

Tag (8 bits)	Set (6 bits)	Word (2 bits)

Processing:

- Model your cache as an array of cache lines. Each cache line should consist of:
    - A TAG field containing a single 8-bit value
    - A VALID field containing a single true/false value
    - A LRU field containing a single true/false value
    - A DATA field containing an array of 4 bytes representing a cache line
- Model your main memory as an array of bytes as stated above.
- You can use addresses provided in the test file to obtain the correct memory array indexes.
- You may assume that the addressable unit is a byte, the replacement algorithm is LRU.
- You should modularize your program properly.

Output:

Output should consist of:

- A section which identifies the parameters of the cache and memory being modeled. At a minimum it should list the number of addressable units per line (or block), the number of sets, and the number of blocks. It should also state that the cache is 2-way set associative with 1 byte words.
- A section detailing each read as it is performed. For each read processed, you should produce a sub-section of output containing:
    - The main memory location being read from and the value being read
    - If a cache miss, print "not in cache". Otherwise print "in cache"
    - The set number and line number being read
    - The data sent to the CPU for this address

Extra marks will be given for implementing a replacement policy where write cycles are possible and where the caches operate a write back policy. In particular consider the replacement policy using a combination of the Modified and LRU flag and the penalty involved when replacing a cache line that is marked modified and hence requires a cache flush and write back to DRAM.

Submit a formal report on your work which should consist of:

1. A background discussion of cache structures and their operation,
2. A description of your particular implementation of the simulator with flowcharts etc. along with
3. A copy of your code which should be heavily commented.

Dr. R. Lynch