Assignment on Cache Memory

CS-201P

Date of evaluation May 16, 2019

1. In this assignment you'll build a simple simulation of a CPU with one register, one levels of cache, and a main memory. You'll run your simulation on a small program of load instructions, moving values from main memory into the register, and determine the total number of hit occurred while referencing memory. Address of load register is given as separate tex file

You can complete this assignment in C++, C, or any other programming language you are familiar with (Java, Python, R, etc.)

Cache Size: 1024, 2048, 4096, 8192, bytes
Block size: 16 bytes
Mapping function: Direct Mapped, 4-way set associative

Replacement Policy: LRU- Require updating the time-stamp of a block every time it is accessed.◦ FIFO- Require updating the time-stamp of a block when it is loaded into cache the first time. The simulator should keep track of cache misses for each combination of size, associativity, and replacement policy.

Considering a sequence of hexadecimal addresses (given with this assignment), simulator executes a read from the cache to 1) check if it is contained in the cache. 2) If not, the number of misses are incremented and the cache is updated using the replacement policies. You are provided with a large file of address traces

The simulator should output results in a table which contain the miss ratio for each combination of size and mapping function for each replacement policy.

Direct mapping				4-way set associative		
Cache Size	1024	2048	4096	1024	2048	4096
LRU						
FIFO						

Marks will be awarded based on the uniqueness of your implementation. Repeat the process if size of block is 32 bytes. Assume that each memory is byte addressable.

2. Finally, find the size and mapping function of your laptops cache:

L1-D, L1-I, L2 and L3 cache