CS 2410 Project 2 Report
By: Xiaoyu Ge, Longhao Li, Yubo Feng

Introduction:

In this project we designed a cache simulator, which can evaluate both MSI and MESI multilevel snooping cache coherence with fully configurable perimeters.

Design:

In our system we support following parameters: **P** (number of cores), **n1** (size of L1 cache), **n2** (size of L2 cache), **k** (size if each block), **a1** (L1 cache associativity), **a2** (L2 cache associativity), **B** (number of banks), **d2** (L2 hit time) and **dm** (L2 miss penalty).

Our project is developed using Java programming language and follows object-oriented principles. For our project we've defined following classes:

Bank.java:

Bank class contains the real content of L2 cache. Since in our design L2 cache is formed using multi Banks. So there is no single place in our simulator represents the entire L2 cache. Also in our system the L2 block indices are interleaved among the L2 banks, which makes the bandwidths of our system much higher.

And for simulate the L2 cache we defined an inner class for each bank called "L2Block". The data structure for store the L2 cache is a double array list, where the inner array list represents a set of L2 block. And the outer array list represents the concentration of L2 data sets.

Bus.java:

Bus.java class is mainly responsible for the control the bus translation. This class is in charge the entire bus transition. It coordinates both L2 to L1 bus transitions and L1 to L2 bus transitions, as well as snoopy bus transitions.

CacheSimulator.java:

This is the main method and main engine of our project. Basicly what CacheSimulator class does is that it will first read the config file and then fetch all commends from specified trace file to each related cores. Each core will have a list contain all commends assigned to this core. And CacheSimulator will keep track of the global cycle, in each it will allow remaining commends to issue and execute base on the situation.

Command.java:

For each command we received, we will create a command class for that command. This class will simulator all the actions that specific command will take during its' execution.

Core.java:

For each core in our system, there is a core class. Which represents that core, it contains many necessary variables, which could be very helpful for reporting result and comparative analysis (for example L1 hit rate and L1 miss rate and etc.). And most importantly each core contains an L1Cache class, which simulates the L1 Cache for that core.

L1Cache.java:

The L1 cache class is used for simulate the L1 cache in our system. For each L1Cache class there is an inner class called L1Block, which represents a L1 block. The data structure for store L1 block in L1 cache is also a double array list just like the one in the bank.java file, therefore again the inner array list represents a set of L1 block. And the outer array list represents the concentration of L1 data sets.

L2Cache.java:

As we stated in the description of Bank class, nothing in our system represents the entire L2 Cache. So the responsibility of L2 cache class is to just provide the unified access interface for L2 cache, and make data block interleaved among all L2 banks.

Comparative Analysis:

Trace1.txt (MSI Mode)

* MSI	Normal	Double number of L2 banks	Double size of L2	Double associativity of L2	Double associativity of L1
Total cycle	89469	88440	89469	89469	89469
L1 hit rate	0.9961335575385342	0.996139043799565	0.9961335575385342	0.9961335575385342	0.9961363006690496
L1 miss rate	0.0038746718530120947	0.0038746718530120947	0.0038746718530120947	0.0038746718530120947	0.0038719287224966877
L2 hit rate	0.031216743526073076	0.029129662522202487	0.031216743526073076	0.031216743526073076	0.03052893148739794
L2 miss rate	0.9563675062078751	0.9577264653641208	0.9563675062078751	0.9563675062078751	0.957046503372382
Miss penalty	187.35651917914817	179.53151340978002	187.35651917914817	187.35651917914817	188.48088060954407

The doubling of numbers of L2 banks will affect most of the data. The most important number is that total cycle dropped a little bit. But the doubling of the size of L2 and doubling associativity of L2 didn’t affect the data. It may due to the very high L1 hit rate. L2 didn’t affect a lot. However, the doubling of associativity of L1 changed the data. L1 hit rate slight higher than others. It is a reasonable result.

Trace1.txt (MESI Mode)

* MESI	Normal	Double number of L2 banks	Double size of L2	Double associativity of L2	Double associativity of L1
Total cycle	89531	88569	89531	89531	89531
L1 hit	0.9962144	0.9962144798	0.9962144798	0.9962144798	0.9962144798

rate	798887387	887387	887387	887387	887387
L1 miss rate	0.0038746718530120947	0.0038746718530120947	0.0038746718530120947	0.0038746718530120947	0.0038746718530120947
L2 hit rate	3.623188405797101 5E-4	0.0	3.6231884057971015E-4	3.6231884057971015E-4	rate3.6231884057971015E -4
L2 miss rate	0.0014492753623188406	0.0025362318840579 71	0.0014492753623188406	0.0014492753623188406	0.0014492753623188406
Miss penalty	272.1789639641164	272.1789639641164	272.1789639641164	272.1789639641164	272.1789639641164