CS203 Fall ’17 – Lab 4 Write-up

Team: Jingnan Cao(861308788), Yue Wang(861306322)

1. How to compile and run

* Our program is written in python, needless to compile.
* To run the program, use

./Lab4.py [cache size in KB] [block size in B] [number of ways] [trace file path]

* Cache size, block size and number of ways should be power of two.
* As for ways, 1 means direct, 2 means 2-ways and so on. Especially, 0 means fully associative.
* For example, if we want to set 512KB cache size, 16B block size, using 4-way associative with trace file gcc-10K.memtrace, we can use the command

./Lab4.py 512 16 4 gcc-10K.memtrace

2. Print out

1. 4K cache size, 16B block size, 4-way associative with file gcc-10K.memtrace

casesensiti (master \*):CS203Lab4 $ ./Lab4.py 4 16 4 gcc-10K.memtrace

miss rate: 11.98%

hit rate: 88.02%

number of sets: 64

ways: 4

number of address bits for tag: 22

number of address bits for index: 6

number of address bits for offset: 4

1. 4K cache size, 16B block size, fully associative with file gcc-10K.memtrace

casesensiti (master \*):CS203Lab4 $ ./Lab4.py 4 16 0 gcc-10K.memtrace

miss rate: 10.94%

hit rate: 89.06%

number of sets: 1

ways: 0

number of address bits for tag: 28

number of address bits for index: 0

number of address bits for offset: 4

1. 4K cache size, 16B block size, 4-way associative with file gcc-1M.memtrace

casesensiti (master \*):CS203Lab4 $ ./Lab4.py 4 16 4 gcc-1M.memtrace

miss rate: 10.0154609591%

hit rate: 89.9845390409%

number of sets: 64

ways: 4

number of address bits for tag: 22

number of address bits for index: 6

number of address bits for offset: 4

1. 4K cache size, 16B block size, fully associative with file gcc-1M.memtrace

casesensiti (master \*):CS203Lab4 $ ./Lab4.py 4 16 0 gcc-1M.memtrace

miss rate: 9.14831979198%

hit rate: 90.851680208%

number of sets: 1

ways: 0

number of address bits for tag: 28

number of address bits for index: 0

number of address bits for offset: 4