

# Computer Organization

## Lab 6: Cache Simulator

Due: 2022/7/6

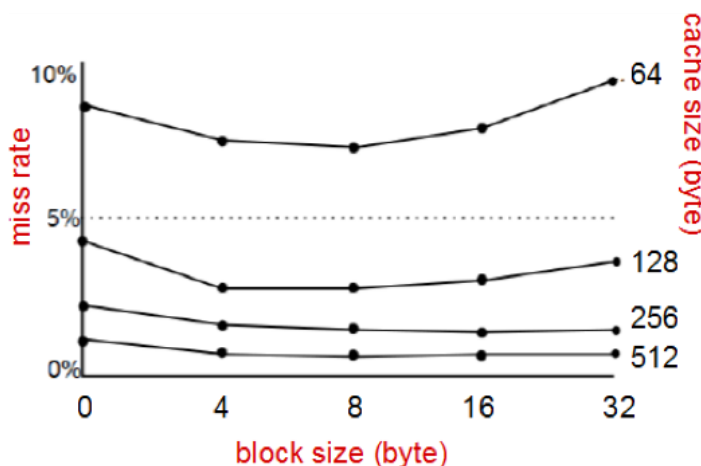
### 1. Goal

Cache performance is important for system performance. In this lab, you are demanded to simulate cache behaviors by C/C++ style cache simulators. By this training, you will understand the performance difference between different cache architectures.

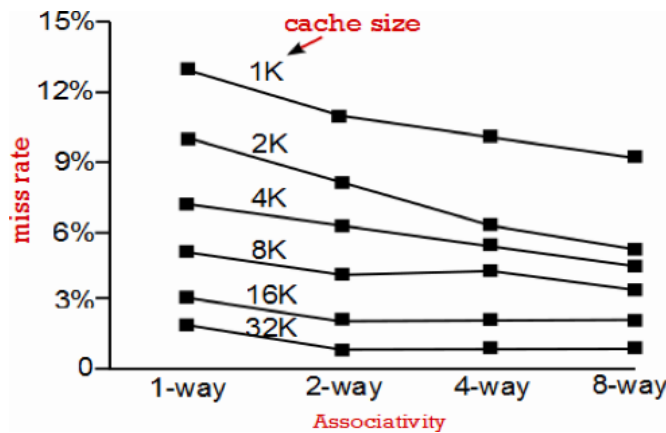
### 2. Problem

In this problem, you have to implement an n-way set-associative cache simulator (LRU replacement policy). Inputting the file Trace.txt that is the memory trace from a benchmark to the simulator. We will supply the direct\_map\_cache.cpp file, you can refer to this and implement your program.

- Fix the associativity on 1 (direct\_map\_cache), and then observe the difference when adapting the cache size and block size. Please draw a graph as the following example and describe the reason of rise and fall of the lines in the report.



- Fix the block size on 32 (byte), and then observe the difference when adapting the cache size and associativity. Please draw a graph as the following example and describe the reason of rise and fall of the lines in the report.



### 3. Input/Output

- Input
  - a. memory trace file (Trace.txt)
  - b. block size (16B to 256B)
  - c. cache size (1KB up to 256KB)
  - d. associativity (from direct-mapped to fully associative)
- Output (instruction indexed from 1)
  - a. Miss rate (%)
  - b. Hits instructions
  - c. Misses instructions

### 4. Program Execution Requirement

**Your Program must be executed as follows**

`./cache_simulator -f {case filename} -c {cache size(KB)} -b {block size(B)} -a {associativity}`

Please write a **Makefile**, and ensure that we can compile your source files and generate the cache\_simulator by using “make” command.

### 5. Execution Example

- File: Trace1.txt (byte address)

```
0xbfa437cc
0xbfa437c8
0xbfa437c4
0xbfa437c0
0xbfa437bc
0xbfa437b8
0xbfa437b8
0xbfa43794
```

0xb8088ea8

0xb8088eac

cache size = 1KB, block size = 32 (bytes)

associativity = 2

Inputs:

`./cache_simulator -f Trace1.txt -c 1 -b 32 -a 2`

Sample outputs that print to stdout: (instruction indexed from 1)

Miss rate: 40

Hits instructions: 2,3,4,6,7,10

Misses instructions: 1,5,8,9

## 6. Requirements

- a. Please implement this Lab in C/C++ language.
- b. Please submit your file to E3.
- c. Please compress your report, Makefile, and the code into one single file. The file should be named as: student\_ID.zip (Format must be correct or you will get some penalty)
- d. Files to upload
  - i. code (.cpp/.c/.h)
  - ii. Makefile
  - iii. report (.pdf)

## 7. Grade

- a. **Total:** 100 points (Report 20% / Program 80%)
- b. **Late submission:** No late submission is allowed.
- c. **Score of this Lab is a reference to bias the final score of this course!**