Computer Organization, Spring 2022

Lab6-Cache Simulator

Due: 2022/06/16 23:59

1 Goal

Cache Performance is important for system performance. In order to understand the performance difference between different cache architectures, you are asked to simulate direct mapped and n-way set associative cache behaviors and written in C++ style.

2 Cache implementation

- Direct-mapped cache (60%)

  Implement a direct-mapped cache simulator and named it “direct_mapped_cache.cpp. Please show your output results table

Miss rate		Block size (Byte)				
		16	32	64	128	256
Cache size (Byte)	4k					
	16k					
	64k					
	256k					

▪ Set-associative cache (30%)

Implement an n-way set-associative cache simulator using LRU (Least-Recently Used) with block size = 64 bytes and named it “set_associative_cache.cpp”. LRU is a cache replacement policy that discards the least recently used items first. Take “testcase.txt” as inputs of the simulator and then run it. Please show your output results table

Miss rate		Associativity (Block size:64B)			
		1-way	2-way	4-way	8-way
Cache size (Byte)	1k				
	2k				
	4k				
	8k				
	16k				
	32k				

3 Grade

(1) Direct mapped cache: 60 points

(2) Set associative cache: 30 points

(3) Report: 10 points

(4) Late submission: 30 percent penalty per week

(5) No plagiarism, or you will get 0 point.

4 Hand in format

- $(groupN)_$(studentid1)_$(studentid2).zip or $(groupN)_$(studentid).zip
    1. direct_mapped_cache.cpp
    2. set_associative_cache.cpp
    3. report.pdf

5 Q&A

- Feel free to ask on HackMD if you need
    - Lab6 討論區
- We will not debug for you