CS2630: Computer Organization
Homework 5
Analysis with a cache simulator

Due May 3, 2017, 11:59pm

Instructions: You must submit your files to Assignments > Homework 5 on ICON.

The files to submit are:

- ✓ One hw5.pdf per partnership with the answers to the questions

Goals for this assignment

- Make predictions about cache hit rate for different programs and cache organizations
- Evaluate cache performance using a simulator

Terminology

- cache line == line == block
- line size: number of bytes in a line
- LRU: "least recently used" replacement strategy for cache lines. On a miss on address X, when there is no room left in the set X maps to, remove the line in the set that was least recently accessed.
- fully associative: each line of memory can go in **any** line of the cache
- direct-mapped: each line of memory goes to **exactly one** line in the cache

Assignment

First, go to "HW5 pairs" on ICON and form a group of two people https://uiowa.instructure.com/courses/51389/groups#tab-4048 .

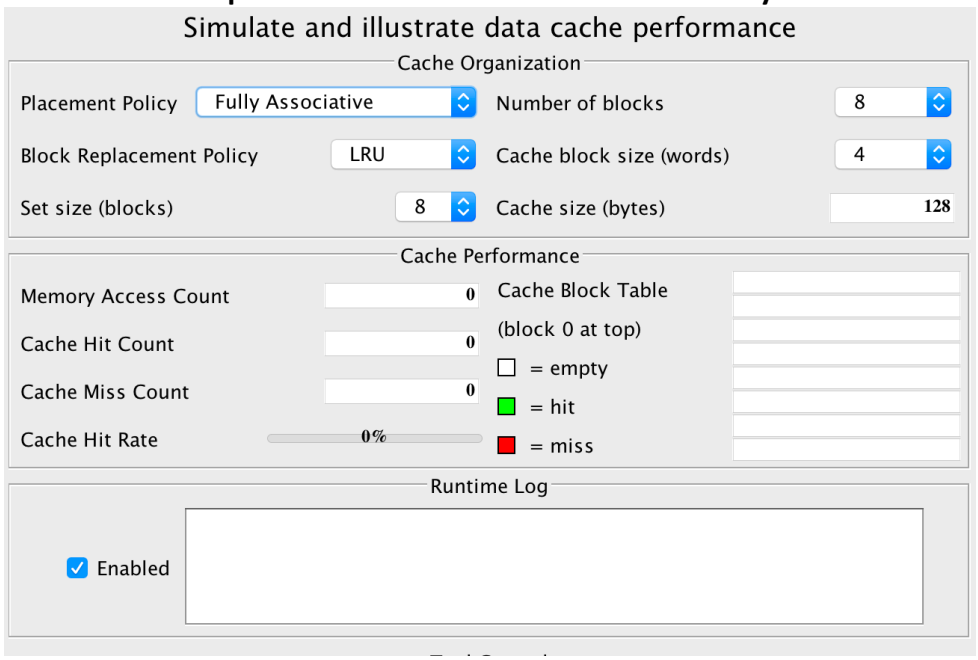
1. Consider the program in `cache_test1.s`. Suppose that the program is run using a fully associative cache with 8 cache lines, a line size of 16 bytes, and LRU.

a) **Predict** the hit rate of the cache when this program runs. You may assume that the syscall doesn't do any memory accesses. You must provide justification for your prediction (in terms of diagrams and calculations).

b) MARS has a cache simulator that can be used to explore the performance of programs on different kinds of caches. The way it works is that the simulator keeps track of the contents of a virtual cache. Each time your MIPS program executes a load or store instruction, the simulator will check if it is a hit and if not then replace a cache line, recording statistics.

Run an experiment with MARs' cache simulator to check your prediction in part (a).

i. Assemble  the program
ii. Tools > Data cache simulator
iii. Configure the cache options to match those given in the problem. **Be aware that block size is specified in words so 4-word lines is 16-byte lines.**



iv. Click Connect to MIPS
v. Ready? Press the Run button  to run your program using the cache simulator.
vi. Does the cache hit rate match your prediction? If not, explain why the hit rate is what the simulator found.

HINT: if you are having trouble analyzing the scenario on paper, the simulator can help!

Reset the simulator 

and reset the MIPS program  and this time

proceed step-by-step  . You'll be able to see what happens in the cache on each cycle by examining the diagram on the right-hand-side and in the Runtime Log.

c) **Predict** what would happen to the hit rate if we changed the cache from fully associative to direct-mapped. Support your answer.

d) **Run an experiment** to check your prediction (**MAKE SURE YOU RESET THE CACHE SIMULATOR first**). Support the result and relate it to the result from (b). Why is there/isn't there a difference?

e) **Predict** what would happen to the hit rate of the fully associative cache if we increase the number of blocks to 16 (doubles the capacity). Support your answer.

f) **Run an experiment** to check your prediction. Support the result and relate it to the result from (b and d). Why is there/isn't there a difference?

g) Of the misses accounted for in "Cache miss count", how many are
Compulsory misses?
Conflict misses?
Capacity misses?

h) **Predict:** If you change the constant `stride` from 4 to 1 will the hit rate of the cache (assuming fully associative/8 lines/16 bytes-per-line) increase, decrease, or stay the same? Support your answer.

i) **Run an experiment** to check your prediction. Support the result if it differs.

j) Is the principle in play in part (h, i) temporal locality or spatial locality? Why?

2. Consider the program `cache_test2.s`. Suppose the cache is fully associative with 8 lines and 16 bytes per line and LRU.

a) **Predict** the hit rate of the cache for this program. Support your prediction.

b) **Run an experiment** to check your prediction. Support the result

c) Of the misses, how many are
Compulsory misses?
Conflict misses?
Capacity misses?

d) How would you reorganize the program so that it has better locality?

3. Consider the program `cache_test3.s`. Suppose the cache is direct-mapped with 4 lines and 8 bytes per line.

a) **Predict** the hit rate for this program. Support your prediction.

b) **Run an experiment** to check your prediction. Support the result if it differs.

c) Of the misses, how many are
Compulsory misses?
Conflict misses?
Capacity misses?