Computer architecture memory design

Assignment 1: Cache: Marks 40

Deadline: Submit by Sep 6, 2024, 11:59pm
Start early. No deadline extensions will be given.

General instructions on what to submit:

- Can be done in groups of 2 or 3 students
- Use any language- C/C++, python, java, Verilog to code
- Submit
  - the code
  - A report showing snapshots
- There will be a demo/viva of the assignment. Note that the marks will be assigned only on satisfactory responses from the group, during the viva
- Submit individual files and not compressed files
- All the files you submit should have the following format: <roll_numbers>_filename
- Only one person in the group will submit the assignment
- If you are getting an error and unable to get the result, submit snapshot of error in the report

What should the report contain?

- Output graphs/tables
- Observations from the experiments

Question –Caches

Problem statement:

a Design a 4-way set associative cache of size 1024kilobytes. Block size: 4 bytes. Assume a 32 bit address. Figure out how many cache lines you need.

You need not implement a main memory, which means you do not need to implement data in the cache. (Implementing a 2^32 memory will be impossible!)

[Hint: For reporting hit/miss, all you need to check is a tag match, and a valid bit, so there is no need to have data in the cache]

Output: You need to report hit/miss rates of the cache for the input memory trace files (5 traces) provided.

b Vary the cache size from 128kB to 4096 kB and repeat the experiment. Note the change in hit/miss rates and plot a graph of miss rate vs cache size. Plot the miss rates for all traces on the same graph. Do all traces behave the same way?

Grading for a + b:

- The code carries 10 marks.
- The report and observation carries 10 marks.
- The demo/viva carries 5 marks.

c. Keeping the cache size at 1024kB, vary the block size from 1 byte to 128 bytes. Note that the number of cache lines will reduce, if you increase the block size keeping the cache size same. Repeat the experiment for all the trace files. Plot the miss rate vs block size. What is the observation? Do all traces behave the same way?

d. Vary the associativity from 1-way to 64-way, for a fixed cache size of 1024kB, and plot the variation of hit rates vs associativity. What is the observation? Why do some traces behave differently?

Grading for c+d:

- The code/experiment carries 5 marks.
- The report and observation carries 10 marks.
- The demo/viva carries 5 marks.

Inputs to your code:

Use the memory trace files at this location

https://cseweb.ucsd.edu/classes/fa07/cse240a/proj1-traces.tar.gz as

input.

The trace file will specify all the memory accesses/addresses that occur in a certain program. Each line in the trace file will specify a new memory reference and has the

following fields:

- Access Type: A single character indicating whether the access is a load ('l') or a store ('s').

You can ignore this field. For reporting hit/miss, it does not matter whether it is a Load/Store

- Address: A 32-bit integer (in unsigned hexadecimal format) specifying the memory address that is being accessed. This is the only field you need.

- Instructions since last memory access: Indicates the number of instructions of any type that executed between since the last memory access (i.e. the one on the previous line in the trace). For example, if the 5th and 10th instructions in the program's execution are loads, and there are no memory operations between them, then the trace line for with the second load has "4" for this field. You can safely ignore this field

Output from the code:

- Hit rates or miss rates for all the 5 traces for the different experiments can be reported in an excel or in the form of a table/graph in the report.

- Submit the graphs/table/excel with your observations in the report along with the code