Assignment 2
Cache Evaluation

Report due: Friday Week 9

Aim:

When designing a processing system it is important to be able to evaluate differing design decisions. In this assignment you are to run write a simulator to help you evaluate the effects of different structures of cache on a system's performance. Your report should explain the result you obtained and be able to make recommendations of cache settings given the variations in the programs used.

Assumed perquisite knowledge:

- A knowledge of how to program in a high level language with particular emphasis on integer bit manipulation.

Method:

Given a set of design parameters below, expand your MIPS matrix multiplication algorithm to a more general case algorithm that can run for various sized matrices. Your algorithm should run for matrix sizes between 2 by 2 to 256 by 256.

Write a simulator for the behaviour of the data cache and evaluate the CPU time of each matrix NxN matrix multiplication program from N=2 to N=256. Your simulator should calculate the execution time required for your programs. You should vary the cache size and architecture. The parameters you should iterate through of your cache and RAM are below:

Cache Specification:

You should simulate 12 different cache configurations, that is, all combinations of the specifications shown below.

CPU clock speed: 1 GHz (1ns period)

Size: 8kB, 32kB, 512kB

Associativity: Direct mapped and 4 way set associative

Words per block: 2, 8

Hit Time:

- 1 CPU clock cycle for 8kB cache
- 2 CPU clock cycle for 32kB cache
- 3 CPU clock cycle for 512kB cache

Miss Time: 2 CPU clock cycles plus time taken to read RAM

DRAM Specification:

RAS Latency: 72 CPU clock cycles

CAS Latency: 24 CPU clock cycles

Bus width: 64 bit

Words per column: 2 (Both words are read with a single CAS access)

Columns per row: 8

DRAM Interleaving: No

To make your job simpler the following assumptions can be made:

- All instructions take a single clock cycle to complete.
- You may treat your MIPs machine as a non-pipelined/ideally pipelined (perfect forwarding) machine.
- Do not worry about the instruction cache. Assumes instructions are always fetched instantly at the start of the cycle.
- You do not have to evaluate the matrix itself, only the time required to compute it on the given system.
- Your matrix code does not have to deal with overflows.
- Assume write through strategy on writes and no write buffer and only restart the processor once all words in the cache block are updated.
- Updates of cache ways can be random and are not needed to be LRU.

Hints and Warnings:

- Pseudo code is available online.
- Parameterize your code so changing cache sizes and types is easy.
- Present your results in a meaningful way. Make it easy to compare execution and relevant data.
- Understand how data is overwritten in a cache. Worst case scenarios are likely to apply.
- Performance of your “processor” is not important for this assignment; you don’t need to optimize your algorithm, keep it simple.
- Both matrices should be in memory in the same order, do not transform either matrix relative to the other unless you complete the transform in your algorithm.
- Keep this code after you have finished your assignment, you will be using it again (or at least you will find it very useful) for assignment 3.
- There is no weekly marks for this report, but it is strongly recommended you use the lab time and demonstrates as a resource for help

Report

There is a report required to be submitted with this assignment. Your report should highlight and explain the interesting features of your results and provide explanations for the different features that become apparent. You should report on:

- Causes of systematic changes in hit rates
- Changes in features caused by changes in cache structure
- Which cache configuration works best for certain matrix sizes

Marking Structure

You will be marked in the following categories:

Cache Simulator Accuracy: 45%

Report: 45%

Code comments and structure: 10%

A summary of the requirements to achieve a high distinction, distinction, credit or pass is in the unit guide.

You should submit your code along with your report.