CS 429/529
Project 2 – Cache Simulator using Gem5

1. Introduction

For Project 2, you will implement the write-back cache models you explored in Project 1 in the gem5 simulation environment and test them using the matrix multiplication benchmark. Your implementation may run on an x86 or ARM CPU in gem5.

Teams

You may solve this homework in teams of two students to increase discussion, participation, and engagement. If you work with another student, only one should submit your solution, including both names in all files.

Program

Instead of using address traces to drive the simulation, as we did in Project 1, gem5 allows us to run real programs.

- Matrix multiplication: You will find the following benchmark program on **`Canvas->Files->Projects->project-2->matmult.c`**. This version will multiply matrices A(K, M) and B(M, N) to produce C(K, N). It initializes A and B with random floats between 1 and 100.

2. Simulations

1. [20] Write-back Cache Class

Develop a gem5 system with x86 or ARM CPU, 4 GB memory, and 256 KB L1 write-back cache (shared instruction and data) components. Make the cache direct-mapped with 32-byte blocks and the memory write-through. Set a hit time of 1 cycle and a miss penalty of 20 cycles. Use the matrix multiplication program to demonstrate that it is working. Then, change the cache model to have separate L1 instruction and L1 data caches, each 256 KB with the same hit time and miss penalty. Again, run the matrix multiplication program to demonstrate that it is working.

2. [20] Write-back Instruction and Data Caches

Change the cache model to have separate L1 instruction and data caches, each with 256 KB, the same hit time, and miss penalty. Again, run the matrix multiplication program to demonstrate that it is working.

3. [20] Write-back Instruction and Data Caches

Conduct a study with the gem5 system above (with both cache models), varying the set associativity: 1, 2, 4, 8, 16, and 32. Produce at least one plot of the results and discuss any insights from the experiments.

This file may change in response to your questions/suggestions.

4. [10] Write-back Instruction and Data Caches

Develop a gem5 system with x86 or ARM CPU, 4 GB memory, *x* KB L1-I cache, y KB L1-D cache (each 2-way set-associative), and *y* MB L2 shared cache (4-way set-associative). Initially, set *x=64 KB*, *y=128 KB*, and *2 MB*. Set an L1 hit time of 1 cycle, an L2 hit time of 10 cycles, and an L2 miss penalty of 100 cycles. Run the matrix multiplication benchmark to demonstrate that it is working. Make a few variations on *x*, *y*, and *z* (minimum 8) and show the results. Produce at least one plot of the results and discuss any insights from the experiments.

3. [20] Report

Please produce a report containing a brief discussion of each section and results generated from the gem5 runs.

3. Submission

You will submit your project report, code, and other files on Canvas. Please submit your report in PDF. Please package your code in a TAR or ZIP file. If you want to provide additional files, please package them in a separate TAR or ZIP file.