Computer Architecture

Project 1

Objective: Become familiar with gem5 and learn implications of different cache configurations

In this project, you will be required to use gem5 processor simulator, which is popular in academia. You will do experiments with different benchmarks and will learn how to simulate the performance on gem5. For various benchmarks, you will analyze impact of different cache parameters on performance metrics. To learn how to download, build and use gem5, please refer to the supporting resources.

Note: We are using Gem5 to simulate the ALPHA processor. If you followed all steps in the Getting Started document, this should already be configured.

Benchmarks: Performance based system-design has made benchmarking a crucial aspect of the design process. Different benchmarks target specific areas of the computation. For example, widely used SPEC (Standard Performance Evaluation Corporation) CPU benchmarks characterize workloads for general purpose computers. They are used to evaluate the efficacy of different microarchitecture design aspects such as different data or instruction-level parallelism techniques, sophisticated caching or branch-prediction techniques etc. For this project, we will use the benchmarks from MiBench suite consisting benchmarks for embedded processors. Mainly, we will deal with networking benchmark dijkstra, binary of which is provided to you. You may also simulate other benchmarks such as FFT, qsort or basicmath on gem5 and analyze the performance statistics.

CPU Model: Gem5 can simulate several CPU models. Please refer to http://www.m5sim.org/CPU_Models to learn more about them. AtomicSimpleCPU model is a functional simulator that uses atomic memory accesses, and can be used to profile instructions, and collect simulation statistics. However, in this project, we will use TimingSimpleCPU model which uses timing memory accesses and models the memory accesses in greater details – which is needed here while dealing with different cache configurations.

How to change the cache configurations while running gem5?

Use the following command line to get help on the options to make changes to the cache architecture configuration:

For example, to simulate the execution of dijkstra benchmark on ALPHA processor with the cache specification such as direct-mapped L1 data cache of 1 kB, 16-way set associative L2 data cache of size 16 kB, L1 instruction cache of 2 kB etc., we can run following command:

```
> ./build/ALPHA/gem5.opt ./configs/example/se.py --cpu-
type=TimingSimpleCPU --caches  --l1d_size=1kB --l1d_assoc=1 --
l2_size=16kB --l2_assoc=16 --l1i_size=2kB --cacheline_size=16 --
l1i_assoc=1  --l2cache --num-l2caches=1 -c
./benchmarks/dijkstra/dijkstra_small -o
./benchmarks/dijkstra/input.dat
```

To understand what each of the above processor parameters represent, please use the script:

> build/ALPHA/gem5.opt configs/example/se.py -h

Things to do before the project:

- In this project, you will analyze the performance statistics for different cache parameters when a benchmark executes on ALPHA processor simulated in gem5. So, it is often easy to create a script file to run your configurations. It will be one-script file, which has all the configuration loaded that you need. So, executing this script file is enough to generate the statistics for these various configurations.

  $> vi run.sh

- Your script file should start with:

  #!/bin/sh

- You can have your configuration loaded in the script file. An example is given below.

```
./build/ALPHA/gem5.opt -d ./dijsktra-Conf1 ./configs/example/se.py --
cpu-type=TimingSimpleCPU --caches --l1d_size=1kB --l1d_assoc=1 --
cacheline_size=16 --l2cache --num-l2caches=1 --l2_size=16kB --
l2_assoc=16 --l1i_size=2kB --l1i_assoc=1 -c
./benchmarks/dijkstra/dijkstra_small -o
./benchmarks/dijkstra/input.dat
```

Usually, executing gem5 without any option stores the output files in the folder m5out. However, **-d** command redirects your stats.txt to a directory called dijsktra-Conf1 (which is created during runtime). To know more about how to save your stats in your directory rather than m5out, explore:

> build/ALPHA/gem5.opt -h

So, make the script file with all your configurations one after another. Then press ESC and (type **:wq)** to save your script. Your script file is now ready and you can execute it.

Problem 1 [5 Points]: Impact of Cache Size on Execution Time and Miss Rate

Caches are vital to computer architecture. In this problem, we analyze the impact of increasing size of L1 data cache. Run the dijkstra benchmark on gem5 with following cache configurations. You can use the command alike the one described before. Please use TimingSimpleCPU model to obtain greater memory access details.

No. of configuration	l1d_size (kB)	Associativity (l1d_assoc)	Cache line size
1	1	1	16
2	2	1	16
3	4	1	16
4	8	1	16

Please note that for each of these configurations, you are changing ONLY the l1 data cache size. Other parameters such as set associativity of the L1 data cache, L1 instruction cache/L2cache size/associativity are same as base configuration (i.e. options are same as --l1d_assoc=1 --cacheline_size=16 --l2cache --num-l2caches=1 --l2_size=16kB --l2_assoc=16 --l1i_size=2kB --l1i_assoc=1).

For example, your first (base) simulation will have L1 data cache size = 1kB, L1 data cache associativity = 1 and cache block size = 16 bits. Your second cache configuration will have L1 data cache size = 2kB, L1 data cache associativity = 1 and cache block size = 16 bits, and so on.

Problem 2 [5 Points]: Impact of Cache Associativity on Execution Time and Miss Rate

Direct mapped caches may suffer from a lot of interference, and are very susceptible to the data placement in memory, most caches are set associative. In this problem, you will examine the impact of the cache associativity on performance. Run the dijkstra benchmark on gem5 with following cache configurations.

No. of configuration	l1d_size (kB)	Associativity (l1d_assoc)	Cache line size
1	1	1	16
2	1	2	16
3	1	4	16
4	1	8	16

Here, we are fixing all the parameters of L1 data cache, L2 cache and instruction cache except the set associativity of L1 data cache.

Project Deliverables

1. Your project submission should contain the following:
    a. Two directories, Problem1 and Problem2.
    b. Project1 and Project2 directories should contain four directories dijkstra-Conf1, dijkstra-Conf2 dijkstra-Conf3 and dijkstra-Conf4.
    c. Each of these configuration directories should contain config.ini, config,json, and stats.txt files. The directory structure should be as following:

```
\---Project1-Submission
    +---Problem1
    |   +---dijkstra-Conf1
    |   |       config.ini
    |   |       config.json
    |   |       stats.txt
    |   |
    |   +---dijkstra-Conf2
    |   |       config.ini
    |   |       config.json
    |   |       stats.txt
    |   |
    |   +---dijkstra-Conf3
    |   |       config.ini
    |   |       config.json
    |   |       stats.txt
    |   |
    |   \---dijkstra-Conf4
    |           config.ini
    |           config.json
    |           stats.txt
    |
    \---Problem2
        +---dijkstra-Conf1
        |       config.ini
        |       config.json
        |       stats.txt
        |
        +---dijkstra-Conf2
        |       config.ini
        |       config.json
        |       stats.txt
        |
        +---dijkstra-Conf3
        |       config.ini
        |       config.json
        |       stats.txt
        |
        \---dijkstra-Conf4
                config.ini
                config.json
                stats.txt
```

Submission Instructions

1) Submit single zip file containing all the files. Please follow the naming conventions correctly.

2) In case where you are doing project in a group, only one group member should make submission through blackboard. All the team members of the group will receive the same score.

Rubric

1. **Problem 1 [5 points]**
    a. Correct configuration from the table from config.ini file. [2 points]
    b. Correct stats in the stats.txt file [3 points]
2. **Problem 2 [5 points]**
    a. Correct configuration from the table from config.ini file. [2 points]
    b. Correct stats in the stats.txt file [3 points]