# CSE 520

**Project 2**

**Due Date: February 14, 2018, 11:59 PM**

**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 ARM 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:

**> build/ARM/gem5.opt -h**

For example, to simulate the execution of dijkstra benchmark on ARM 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/ARM/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/ARM/gem5.opt configs/example/se.py -h

#### Automation of Launching Multiple Simulations through a Script:

* In this project, you will analyze the performance statistics for different cache parameters when a benchmark executes on ARM 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 (shell) script file should start with:

## #!/bin/sh

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

./build/ARM/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).
* Make the script file with all your configurations. Then press ESC and (type **:wq)** to save your script and to exit the editor. Your script file is now ready and you can execute it.

**Problem 1 [15 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 bytes. Your second cache configuration will have L1 data cache size = 2kB, L1 data cache associativity = 1 and cache block size = 16 bytes, and so on.

Plot two graphs, first one shows the execution time (sim ticks) for different configurations and the second one shows the overall data cache miss rate for different configurations. The only variable here is L1 data cache size. (10 Points)

Justify the plots by explaining the relation between execution time vs increase in cache sizes and overall data cache miss rate vs increase in cache size. Briefly explain how increase in cache size affects different types of cache misses such as capacity miss, compulsory miss and conflict miss

i.e. how some of them increases or decreases with variation in cache size. (5 points)

**Problem 2 [15 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 | 4 | 1 | 16 |
| 2 | 4 | 2 | 16 |
| 3 | 4 | 4 | 16 |
| 4 | 4 | 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. For these four different cache configurations, plot two graphs – first one shows the execution time (sim ticks) for different configurations and the second one shows the overall data cache miss rate vs varied L1 data cache associativity. (10 Points)

Justify the plots by explaining the relation between execution time vs increase in associativity and overall data cache miss rate vs increase in associativity. Briefly explain how increase in associativity affects different types of cache misses such as capacity miss/compulsory miss/conflict miss. (5 points)

**Problem 3 [10 Points]:** Impact of Block Size on Miss Rate

Now, you will examine the impact of the cache line size on performance. Run the dijkstra benchmark on gem5 with following cache configurations. All parameters are fixed except cache line size. After simulating the benchmark for these configurations, plot a graph showing overall data cache miss rate vs varied cache line size. (5 Points)

|  |  |  |  |
| --- | --- | --- | --- |
| **No. of configuration** | **l1d\_size (kB)** | **Associativity (**l1d\_assoc) | **Cache line size** |
| 1 | 1 | 2 | 8 |
| 2 | 1 | 2 | 16 |
| 3 | 1 | 2 | 32 |
| 4 | 1 | 2 | 64 |
| 5 | 1 | 2 | 128 |

Justify the plot by explaining the relation between overall data cache miss rate vs increase in cache line size. Briefly explain how increase in cache block size affects different types of cache misses such as capacity miss/compulsory miss/conflict miss. (5 points)

**Problem 4 [30 Points]:** Performance vs. Energy Cost Trade-off

In this problem, you will run different cache configurations and will observe the performance improvements over baseline. Baseline is configuration 1 i.e. L1 data cache of size 1 kB, associativity 1 and cache line size is 16. Now, performance may improve with the increase in these cache parameters but such improvement does not come for free. Meaning, they may increase energy consumption. So, we are interested in analyzing the trade-off between performance and energy consumption.

|  |  |  |  |
| --- | --- | --- | --- |
| **No. of configuration** | **l1d\_size (kB)** | **Associativity (l1d\_assoc)** | **Cache line size(byte)** |
| 1-4 | 1 / 2 / 4 / 8 | 1 | 16 |
| 5-8 | 1 / 2 / 4 / 8 | 2 | 16 |
| 9-12 | 1 / 2 / 4 / 8 | 4 | 16 |
| 13-16 | 1 / 2 / 4 / 8 | 8 | 16 |
| 17-20 | 1 / 2 / 4 / 8 | 1 | 32 |
| 21-24 | 1 / 2 / 4 / 8 | 2 | 32 |
| 25-28 | 1 / 2 / 4 / 8 | 4 | 32 |
| 29-32 | 1 / 2 / 4 / 8 | 8 | 32 |

## Cache configuration table

Note: For each of these configurations change either L1 data cache size or the associativity or cache line size. Other parameters such as L1 instruction cache/L2cache size/associativity are always same as baseline (--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 bytes. Your second cache configuration will have L1 data cache size = 2kB, L1 data cache associativity = 1 and cache block size = 16 bytes, and so on.

Name your directories accordingly because all the simulations results are stored by the order in which you specify. You may split your workload with your partner and run simulations in parallel so that you can have your results faster.

For any configuration, we can normalize the execution time (sim ticks) by that of baseline (configuration 1). We can associate a cost for each cache parameter that is used in design of memory hierarchy. So, you will compute cost metric for each configuration.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Cache Size (in kB) | | | | Set Associativity | | | | Cache Line Size | |
| 1 | 2 | 4 | 8 | 1 | 2 | 4 | 8 | 16 | 32 |
| Cost | 1 | 1.5 | 2 | 4 | 1 | 2 | 3 | 6 | 1 | 2 |

For example, for baseline (config 1), cost can be computed as –

cost(1 kB) + cost(associativity=1) + cost(cache line size =16) = 1 + 1 + 1 = 3 units. Similarly, for configuration 10, the cost can be computed as 1.5 + 3 + 1 = 5.5 units.

You will compute the cost in similar manner for all the configurations. Please note that this cost represents the energy consumption and not the price of the processor chip. Tabulate the execution time (sim ticks), cost and cache efficacy.

Now, cache efficacy can be calculated with following equation:

### 1

𝐶𝑎𝑐ℎ𝑒 𝑒𝑓𝑓𝑖𝑐𝑎𝑐𝑦 =

𝑐𝑜𝑠𝑡 × 𝑛𝑜𝑟𝑚𝑎𝑙𝑖𝑧𝑒𝑑 𝑒𝑥𝑒𝑐𝑢𝑡𝑖𝑜𝑛 𝑡𝑖𝑚𝑒2

1. Tabulate four fields – the execution time (sim ticks), normalized execution time (to baseline), cost and cache efficacy for each of the 32 configurations.
2. Plot a graph showing the cache efficacy for different configurations. X axis  Number of configuration.

Y axis  Cache efficacy pf the respective cache-configuration.

Which is the most cost-efficient cache configuration for the given program (has highest cache efficiency)?

#### Problem 5 [10 Points]

* 1. Where are the source files related to the cache implementation located in gem5 directory? Please provide the full path.
  2. If the data needed is not in the cache (cache miss), it needs to be brought in the cache. If cache is already filled, then some block in the cache needs to be evicted. Such eviction mechanism is refereed as cache replacement policy which plays a vital role in improving system performance. A well-known policy is least recently used (aka LRU). Search for such implementation in gem5 and provide the path where source files pertaining to LRU are located? (Hint – You may search within the directory containing cache files.)

**Problem 6[5 points](For 520 students)**

Suppose you were to change the size of L2 to 128kB.

1. If you were to add this configuration to the other L1 cache configurations you’ve run, what L1/L2 configuration combination yields the most efficient cache config combo? You can use the following metric: R = T\*S\*B\*C. T is the runtime in SimTicks, S is the size of the L2 cache, B is the block size, and C is the energy cost of the L1 cache. Assume that since we T,S, B and C to be low as possible, we want R as low as possible. Justify your answer with an explanation of the effect of increasing L2 on performance.
2. What is the effect of increasing the size of L2 on the overall data miss rate, and why?
3. What is the effect of increasing 128kB to 256kB on the runtime, and why?

#### Submission Instructions

1. Write your answers to the questions, and briefly explain required details. Plot needed graphs and/or tables. Submit a single PDF file containing all the answers/plots/tables. Along with reporting output statistics, analyze well these statistics and obtained plots and try to relate it with the concepts.
2. Name the PDF file Project2.pdf. In case where you are doing project in a group, only one group member should make submission through blackboard.
3. There should be no name or ASU ID number of any of the team members in any part of the file or program for the project submission. If a team violates this policy, the score for the project will be ZERO. In short, the submission must be anonymous for a double blinded scoring process.