

## Assignment 1

Indian Institute of Technology Kharagpur

**gem5** is a system simulator that models CPUs at the microarchitecture level and all other associated structures such as caches, memory, interconnect buses, etc. It implements many architectural features that we have studied in this class. The purpose of this assignment is to help you get acquainted with **gem5** and with the methodology for running simulations to estimate the performance of machines vis-à-vis different benchmark programs.

**gem5** allows the simulation of a wide range of CPUs, both functionally (correctness only) and for timing (correctness and efficiency). It supports multiple ISAs, including x86-64. *A tutorial session has already been provided on how to install and setup gem5.* The detailed slides are available on Moodle/ MS Teams. For more information on **gem5**, please visit <http://learning.gem5.org/book/>

**gem5** can simulate CPUs with different configurations (e.g. number of cores, pipeline complexity, cache size...) based on configuration scripts. How to write a sample configuration script has been shown in the tutorial (video available Teams). In this assignment, your task is to configure an **Out-Of-Order** CPU with a list of the microarchitectural parameters (provided below) that reflects the characteristic of a single-core x86 processor and run the provided benchmark program. More precisely, you will have to create different configuration combinations based on the parameters and their values mentioned in this document and run the provided benchmark program using those scripts. Then you will analyse the output statistics of each of these config combinations to select top 10 combinations and finally answer the questions mentioned in the later part of the document.

## Procedure

1. Follow the instructions discussed in the tutorial session to download, build and configure **gem5**. For your own understanding, run the benchmark program on your machine (outside **gem5**). You should also read through the source code because you will need to run the same benchmark from your custom config script.
2. Configure your custom config script to reflect the following fixed parameters:
  - CPU model: Out-of-Order (DerivO3CPU)
  - Caches: L1I, L1D, L2 (set associative)
  - Clock Frequency: 2GHz
  - Memory mode: timing
  - Memory size: 1GB
  - Memory controller: MemCtrl()
  - DRAM type: DDR3\_1600\_8x8()
  - NumberofReorderBuffer: 1
  - l1 tag\_latency: 2
  - l1 data\_latency: 2
  - l1 response\_latency: 2
  - l1 mshrs: 4
  - l1 tgts\_per\_mshr: 20
  - l2 tag\_latency: 20

- l2 data\_latency: 20
  - l2 response\_latency: 20
  - l2 mshrs: 20
  - l2 tgts\_per\_mshr: 12
  - cacheline: 64
3. The following parameters have multiple values. You are required to include all the parameters and test your script with each value of the parameters. For example, if there are  $m$  parameters, each having  $n$  values, you have to run  $n^m$  different configurations to cover all possible combinations. The variable parameters are listed below:
- l1d\_size: 32kB, 64kB
  - l1i\_size: 32kB, 64kB
  - l2\_size: 128kB, 256kB, 512kB
  - l1\_assoc: 2, 4, 8
  - l2\_assoc: 4, 8
  - bp\_type: TournamentBP, BiModeBP, LocalBP
  - ROBEntries: 128, 192
  - numIQEntries: 16, 64
4. Run simulations of the benchmark program using your custom script by changing the values of the parameters mentioned in Point 3. You can also use `gem5/configs/example/se.py` and `gem5/config/common/options.py` with some modifications.
5. Answer the following questions in a PDF document.
- Analyze the `m5out/stats.txt` to extract different statistics for each of the config combinations.
    - (a) Based on the CPI values, which are the top 10 configurations for the benchmark program?
    - (b) Why do you think these combinations of parameters works best for the given benchmark program? You might need to analyse the program source code in order to justify your claims.
    - (c) Provide a graph or plot for each of the top 10 combinations depicting the following -
      - Cycles Per Instruction (CPI).
      - Mispredicted branches detected during execution.
      - Number of branches that were predicted not taken incorrectly.
      - Number of branches that were predicted taken incorrectly.
      - Instructions Per Cycle (IPC).
      - Number of BTB hit percentage.
      - Number of overall miss cycles, miss rate, average overall miss latency.
      - The number of ROB accesses (read and write both).
      - Number of times the LSQ has become full, causing a stall.
      - Number of loads that had data forwarded from stores.
      - Number of times access to memory failed due to the cache being blocked.

## Deliverables

You are required to include the following documents in a compressed folder (zip or tar.gz) titled in the format `<Group_name>_HPCA_Assignment_1`.

1. The custom config script used to run the top 10 parameter combinations. If you have used `se.py` and `options.py`, include those files along with the required commands to execute each of the top 10 combinations.
2. The `m5out/stats.txt` files of the top 10 parameter configurations.
3. A detailed report (in pdf) that contains all the requirement mentioned in Point 5. Also include the names of the group members and their individual contribution to this project.