Lab One: Familiarization with gem5

What is gem5?

gem5 is a modular discrete event driven computer system simulator platform. gem5 is written primarily in C++ and python. It can simulate a complete system with devices and an operating system in full system mode (FS mode), or user space only programs where system services are provided directly by the simulator in syscall emulation mode (SE mode). There are varying levels of support for executing Alpha, ARM, MIPS, Power, SPARC, RISC-V, and 64 bit x86 binaries on CPU models including two simple single CPI models, an out of order model, and an in order pipelined model. A memory system can be flexibly built out of caches and crossbars or the Ruby simulator which provides even more flexible memory system modeling.

Getting Started

Building gem5
https://www.gem5.org/documentation/learning_gem5/part1/building/

Create a simple configuration script
https://www.gem5.org/documentation/learning_gem5/part1/simple_config/

Add Cache
https://www.gem5.org/documentation/learning_gem5/part1/cache_config/

Analyze gem5 statistics
https://www.gem5.org/documentation/learning_gem5/part1/cache_config/

Do this in the context of the typical “helloWorld”.

Write a simple yet non trivial Program

Matrix Matrix multiply [MM] is an important kernel in [almost] all ML/AI workloads. Write a program to implement MM. Run this program [with relatively small input sizes [~100/200] in gem5 and change the CPU model, CPU frequency [600MHz to 3.3 GHz [stepsize 200MHz]], and memory configuration [check what models are present] and describe the changes in performance.

What to submit

Report documenting the performance variations [graphs and tables], MM code and the config file used to generate the results.

Bonus [no extra grade]

Repeat this for different ISA [ARM or RISCV].