

ECE4700J Lab 4 - First Try Architectural Simulations

Xinting Jiang

UM-SJTU JI

evansjiang@sjtu.edu.cn

June 16, 2024

Overview

- 1 Lab Logistics
- 2 Intro to Gem5
 - Intro to Simulators
 - Getting Started with Gem5
 - Customize Your Configurations
- 3 Understand Gem5 Outputs
- 4 Tips for Lab 4

Lab Logistics

Lab 4 Administrivia

- Due time: 23:59 on June 25th (Beijing time)
- Contents
 - Install and build gem5
 - Learn how to configure your own platform in gem5
 - Learn how to do simple cross-compilation
- We highly recommend you use a provided JI server account in VMware Horizon Client, called Gem5.
 - Account: ji+STUDENTID
 - Password: STUDENTID
 - Please have a try ASAP and tell us if you fail to login!!!

Intro to Gem5

Why We Need Architectural Simulators?

For Research in Computer Architecture: How to implement and test your hardware design?

- Use Self-Produced Real Chips (ASIC: application specific integrated circuit) (e.g. Google's TPU)
 1. Chip Fabrication is extremely expensive
 2. Very long Development Cycle
 3. Designing ASIC is too complicated
- Use Programmable Devices. E.g. FPGA
 1. +: Enables people to do hardware designs with low cost
 2. +: Shorter Development Cycle (Is it really short?)
 3. -: Limited capacity and performance. Imagine you really implement a complex CPU on FPGA... On that CPU, even boosting Linux may take minutes.
 4. -: Poor software environment/tools (e.g. Vivado (- -)||)

Why We Need Architectural Simulators?

Use Software Simulators!

1. They can model any hardware you want
2. Very good scalability
3. Short Development Cycle
4. Good Environment Support (Since it's software! E.g. You can use printf/gdb to debug!)

However, at last they are still not real hardware. There are things that they cannot model.

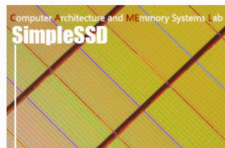
Nowadays, developing better simulators / framework for FPGA / Language for programming hardware are all hot topics in computer architecture research.

Simulator Examples

Processor Simulators



Memory Simulators



GPU Simulators



Link:

1. <https://www.gem5.org/>
2. <https://docs.simplessd.org/en/v2.0.12/>
3. <https://accel-sim.github.io/>

What is Gem5?

Michigan m5 + Wisconsin GEMS = gem5

“The gem5 simulator is a modular platform for computer-system architecture research, encompassing system-level architecture as well as processor microarchitecture.”

A open-source tool!

Build Gem5

In your terminal, after you make sure all the dependencies have been installed, run

```
git clone https://gem5.googlesource.com/public/gem5
cd gem5
scons build/X86/gem5.opt -j 4
```

And now wait...

Build Gem5 (Cont.)

For instruction `scons build/X86/gem5.opt -j 4`

- `scons`: the build system that gem5 uses (like make), See <http://scons.org/>
- `build/X86/gem5.opt`: “parameter” passed to `scons`. gem5’s Sconscript interprets this. Also, the patch to the gem5 executable
- `X86`: specifies the default build options. See `build_opts/*`
- `opt`: version of executable to compile (one of debug, opt, perf, fast)

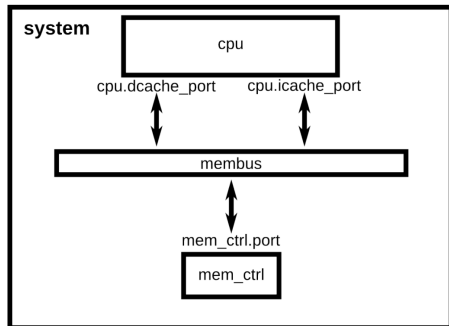
Gem5 SimObject

- gem5 consists of “SimObjects”
- Most C++ objects in gem5 inherit from class SimObject
- Represent physicalsystem components (e.g. cache)

Gem5 User Interfaces

- gem5 completely controlled by Python scripts
- Scripts define system to model
- All (C++) SimObjects exposed to Python

Simple Config Script



- Single CPU connected to a memory bus
- provided script in gem5:
`configs/learning_gem5/part1/simple.py`

Port Interface

```
system.cpu.icache_port = system.membus.slave
system.cpu.dcache_port = system.membus.slave
...
system.mem_ctrl.port = system.membus.master
```

Use “=” sign in Python to connect a master port and a slave port from different devices.

Running Gem5

```
build/X86/gem5.opt configs/tutorial/simple.py
```

- `build/X86/gem5.opt`: the `gem5` binary to run
- `configs/.../simple.py`: the configuration script (config script)

Different Simulation Modes

- SE mode: system emulation. Emulates the operating system syscalls. No OS runs.
- FS mode: full system. Runs a full OS as if gem5 is a “bare metal” system.
- In this lab, we use the SE mode.

Understand Gem5 Outputs

Gem5 Outputs

After you running a gem5 simulation,

```
>ls m5out/  
config.ini config.json stats.txt
```

- `config.ini`: Dumps all of the parameters of all SimObjects. This shows exactly what you simulated.
- `config.json`: Same as `config.ini`, but in `txt` format.
- `stats.txt`: Detailed statistic output. Each SimObject defines and updates statistics. They are printed here at the end of simulation.

Tips for Lab 4

Tips for Lab 4

- Very easy but you need to read the instructions in the manual carefully
- Installation and building of gem5 may take longer time so you still need to start early
- If you meet any “mysterious” bugs, please post the screenshot error message and describe your condition neatly.

Reference

[1] Haoyang Zhang, ECE4700J SU 2022 Lab6

Thank you!