Lab 1: Building X86 and ARM Architectures in gem5

EEL 5734 – Fall 2022

Introduction:

In this lab, you will follow a gem5.com tutorial to set up a gem5 development environment and build X86 and ARM architectures in gem5. This X86 architecture will be used for the subsequent labs.

Part 1 – For Windows machines, download a virtual machine and install Ubuntu

The gem5 development environment is set up in a Linux environment. Thus, if you are using a Windows machine, you need to set up a virtual machine to run Linux. If you are using a machine running Linux, then **you can skip Part 1 and go directly to Part 2 of this lab**.

Download and install a virtual machine (VM) on your Windows machine: You can download free virtualization solutions such as VMware Player or VirtualBox. Follow the manufacturer's instruction in installing the VM . Note: I set up my VM using VMware Player. In doing so, I used all the default values except:

- I changed "maximum disk size" to 50GB (default maximum disk size 20GB).
- I changed the "memory for this virtual machine" to 8GB (default was 4GB). Note that you have to click the "Customize Hardware…" button (before the "Finish" button) to change the memory to 8GB. Also, if you still encounter errors, you may have to use more than 8GB (e.g., 12GB).

Also note that you are able to create the VM using just the default values. However, when I tried to build a gem5/X86 architecture (Part 2), I encountered errors.

Now follow the instruction to install a Ubuntu operating system onto the VM. See the gem5 requirements for the currently supported versions of Ubuntu:

https://www.gem5.org/documentation/general_docs/building#dependencies

Part 2 – Building a gem5/X86 architecture

In part 2, you will follow the tutorial ("Building gem5") from the gem5.com website to install all the necessary applications and build a gem5/X86 architecture:

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

When you reach the build command line: `` python3 `which scons` build/X86/gem5.opt -j9 ``

- cd to the ~/gem5/ directory
- Change -j9 to -jx, where x = number of cores in your computer + 1. In my case, my command is the following because I have 6 cores:
  python3 `which scons` build/X86/gem5.opt -j7

When compilation is finished you should have a working gem5 executable (gem5.opt) in the ~/gem5/build/X86 directory.

Part 3 – Building a gem5/ARM architecture

If you have successfully built the gem5/X86 architecture in Part 2, then building a gem5/ARM architecture is very easy. Simply go back to the ~/gem5/ directory and type in the following command:

```
python3 `which scons` build/ARM/gem5.opt -j7
```

(again, the -j number should be #cores + 1)

When finished, verify that you have a working gem5 executable (gem5.opt) in the ~/gem5/build/ARM directory.

SUBMISSION INSTRUCTIONS

Turn in one pdf file with the following naming convention:

- **LastNameFirstNameLab1.pdf** ex: LamHermLab1.pdf

The **pdf file** contains screenshots of the outputs copied/pasted from your Ubuntu terminal screen, after you executed each of the following commands:

1. python3 `which scons` build/X86/gem5.opt -j7
2. python3 `which scons` build/ARM/gem5.opt -j7
3. After you have completed all parts of Lab 1, go to the **~/gem5/build/X86** directory and type a **ls -l** command. It should show the executable **gem5.opt**.
4. Do the same (**ls -l**) at the **~/gem5/build/ARM** directory.
5. **Post-lab question:** Does gem5 support the RISC-V CPU? If yes, then what is the command to build it? (You will have to do some searching to find the answer.)