Computer Architecture
Lab Assignment Logistics

Institute of Computer Science and Engineering
National Yang Ming Chiao Tung University

revision: 9-2-2024

Welcome to the course! This document outlines the fundamental guidelines for maintaining academic integrity in the lab and offers instructions for setting up the environment for the upcoming labs in this course.

1 Lab Academic Integrity

Students are expressly prohibited from sharing their code with anyone who is not a member of their assigned group or a course staff member. Furthermore, students are strictly forbidden from referencing or replicating lab reports from previous iterations of the course. All submitted lab reports must be independently developed and should accurately reflect the students' own understanding of the lab assignment.

While not required, students have the option, though not the obligation, to utilize generative AI in this course. Should you choose to use generative AI, it is essential to disclose this in your lab reports. However, please be mindful that materials produced by such tools may be inaccurate, incomplete, or otherwise flawed. It is imperative that you exercise your best judgment.

Any sharing of code or copying from previous lab reports will be considered a violation of course regulations. Additionally, the distribution of any lab materials in public forums (e.g., GitHub) is strictly prohibited and will also be regarded as a breach of course regulations. Students found responsible for such actions will face severe penalties, which will likely result in failing the course.

2 Lab Late Submission Policy

Timely submission of lab assignments is essential for staying aligned with the course material. The penalties for late submissions are as follows:

- **Late by up to one week:** Lab assignments submitted within one week after the deadline will be eligible for a maximum of 80% of the original score.
- **Late until the end of the semester:** Lab assignments submitted more than one week late but before the end of the semester will be eligible for a maximum of 60% of the original score.

Please plan accordingly to avoid late penalties.

3 Setting Up Environment

- Use `WSL2` for Windows
  1. Install `WSL2` : Open `Windows Powershell` and type

```
% wsl --install -d Ubuntu-22.04
```

2. Install basic tools including `make`, `iverilog` and `GTKWave` in `wsl` with following commands :

   ```
   % sudo apt-get update
   % sudo apt-get install build-essential
   % sudo apt-get install iverilog
   % sudo apt-get install gtkwave
   ```

3. Install dependencies for `RISC-V GNU` toolchain :

   ```
   % sudo apt-get install autoconf automake autotools-dev gawk bc
   % sudo apt-get install libmpc-dev libmpfr-dev libgmp-dev flex
   % sudo apt-get install libexpat-dev libtool patchutils git curl
   % sudo apt-get install texinfo bison gperf zlib1g-dev python2
   ```

4. Install `RISC-V GNU` toolchain :

   ```
   % git clone https://github.com/riscv/riscv-gnu-toolchain
   % cd riscv-gnu-toolchain
   % ./configure --prefix=/opt/riscv --with-arch=rv32imzicsr
   % sudo make -j$(nproc)
   % sudo make install
   % export PATH=/opt/riscv/bin:$PATH
   % source ~/.bashrc
   ```

   Note : `-j$(nproc)` means to use all available processors to run the `make` command.

- Use `Docker` for MacOS

1. Download and install `Docker` :

   https://docs.docker.com/desktop/install/mac-install/

2. Pull image :

   ```
   % docker pull tinggjung/ca_env:v0
   ```

3. Run image :

   ```
   % docker run -it tinggjung/ca_env:v0
   ```

4. Install `Homebrew` :

   ```
   % /bin/bash -c "$(curl -fsSL https://raw.github\
   usercontent.com/Homebrew/install/HEAD/install.sh)"
   ```

5. Install `GTKWave` :

   ```
   % brew install --HEAD randomplum/gtkwave/gtkwave
   ```

6. Copy the VCD file to the host to use with `GTKWave` :

   ```
   % docker cp [container ID]:[path to file] [path on host]
   ```

   Note : You can mount the volume from the container to the host to skip this step.