

Computer Architecture

Lab Assignment Logistics

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

revision: 9-1-2025

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 strictly prohibited from sharing their code with anyone outside their assigned group or course staff. Additionally, referencing or replicating lab reports or materials from previous iterations of the course is forbidden. All submitted lab reports must be independently developed and must accurately reflect the students' own understanding of the lab assignment.

Students may optionally use generative AI tools in this course. Be cautious, as AI-generated content may be inaccurate, incomplete, or flawed. While generative AI can assist in learning, students must critically evaluate and adapt AI outputs to ensure the work reflects their own understanding. Regardless of the source, **students are fully responsible for ensuring the originality and accuracy of their submissions**. Submitting work identified as plagiarized or **insufficiently original—whether generated by AI or otherwise**—will be considered a serious violation of academic integrity and will result in penalties.

Students are also reminded that any form of **sharing lab materials or reports in public forums (e.g., GitHub) or with unauthorized individuals is strictly prohibited**. Any instance of plagiarism, unauthorized sharing, or other breaches of academic integrity will result in severe penalties, **up to and including failure of the course**.

2 Setting Up Environment

2.1 Windows / Linux

Use WSL for Windows

1. Install WSL2 : Open Windows Powershell and type

```
% wsl --install -d Ubuntu-24.04
% sudo apt update
```

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

```
% wsl
% sudo apt update
% sudo apt install build-essential autoconf
% sudo apt install iverilog
```

3. Install GTKWave:

```
% cd ~
% wget https://archive.ubuntu.com/ubuntu/pool/universe/g/gtkwave\
/gtkwave_3.3.104-2build1_amd64.deb
% sudo apt install ./gtkwave_3.3.104-2build1_amd64.deb
```

4. Install RISC-V GNU Compiler Toolchain:

```
% cd ~
% wget https://github.com/nycu-arclab/riscv32-toolchain/releases/download\
/v14.2.0/riscv32-unknown-elf-gcc-14.2.0-linux.tar.gz
% sudo tar -xzf riscv32-unknown-elf-gcc-14.2.0-linux.tar.gz -C /opt
% echo 'export PATH=/opt/riscv/bin:$PATH' >> ~/.bashrc
% source ~/.bashrc
```

5. Check the iverilog version

```
% iverilog -v
```

If the iverilog installation is correct, the following output will be displayed:

```
Icarus Verilog version 12.0 (stable) ()
Copyright (c) 2000-2021 Stephen Williams (steve@icarus.com)
```

6. Check the gtkwave version:

```
% gtkwave -v
```

If the gtkwave installation is correct, the following output will be displayed:

```
GTKWave Analyzer v3.3.104 (w)1999-2020 BSI
```

7. Check the RISC-V GNU Compiler Toolchain version:

```
% riscv32-unknown-elf-gcc -v
```

If the RISC-V GNU Compiler Toolchain installation is correct, the following output will be displayed:

```
Using built-in specs.
COLLECT_GCC=riscv32-unknown-elf-gcc
COLLECT_LTO_WRAPPER=/opt/riscv/bin/./libexec/gcc/riscv32-unknown-elf/14.2.0/lto-wrapper
Target: riscv32-unknown-elf
Configured with: /home/jack/temp/riscv-gnu-toolchain/gcc/configure --target=riscv32-unknown-elf --prefix=/home/jack/temp/riscv --disable-shared --disable-threads --enable-languages=c,c++ --with-pkgversion= --with-system-zlib --enable-tls --with-newlib --with-sysroot=/home/jack/temp/riscv/riscv32-unknown-elf --with-native-system-header-dir=/include --disable-libmudflap --disable-libssp --disable-libquadmath --disable-libgomp --disable-nls --disable-tm-clone-registry --src=../gcc --disable-multilib --with-abi=ilp32 --with-arch=rv32imzicsr --with-tune=rocket --with-isa-spec=20191213 'CFLAGS_FOR_TARGET=-Os -mcmodel=medlow' 'CXXFLAGS_FOR_TARGET=-Os -mcmodel=medlow'
Thread model: single
Supported LTO compression algorithms: zlib
gcc version 14.2.0 ()
```

2.2 MacOS

1. Install Xcode Command Line Tools:

```
% xcode-select --install
```

2. Install Homebrew iverilog and GTKWave in MacOS with following commands:

```
% /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/
install/HEAD/install.sh)"
% eval "$(/opt/homebrew/bin/brew shellenv)"
% brew install icarus-verilog python3 make
% brew install --HEAD randomplum/gtkwave/gtkwave
```

3. Install RISC-V GNU Compiler Toolchain:

```
% brew install wget
% wget https://github.com/nycu-arclab/riscv32-toolchain/releases/download\
/v14.2.0/riscv32-unknown-elf-gcc-14.2.0-apple-darwin.tar.gz
% sudo tar -xzvf riscv32-unknown-elf-gcc-14.2.0-apple-darwin.tar.gz -C /opt
% echo 'export PATH=/opt/riscv/bin:$PATH' >> ~/.zshrc
% source ~/.zshrc
```

4. Check the iverilog version

```
% iverilog -v
```

If the iverilog installation is correct, the following output will be displayed:

```
Icarus Verilog version 12.0 (stable) ()
Copyright (c) 2000-2021 Stephen Williams (steve@icarus.com)
```

5. Check the gtkwave version:

```
% gtkwave -v
```

If the gtkwave installation is correct, the following output will be displayed:

```
GTKWave Analyzer v3.4.0 (w)1999-2022 BSI
```

6. Check the RISC-V GNU Compiler Toolchain version:

```
% riscv32-unknown-elf-gcc -v
```

If the RISC-V GNU Compiler Toolchain installation is correct, the following output will be displayed:

```
Using built-in specs.
COLLECT_GCC=riscv32-unknown-elf-gcc
COLLECT_LTO_WRAPPER=/opt/riscv/libexec/gcc/riscv32-unknown-elf/14.2.0/lto-wrappe
r
Target: riscv32-unknown-elf
Configured with: /Users/weiluo/Developer/riscv-gnu-toolchain/gcc/configure --tar
get=riscv32-unknown-elf --prefix=/opt/riscv --disable-shared --disable-threads -
-enable-languages=c,c++ --with-pkgversion= --with-system-zlib --enable-tls --wit
h-newlib --with-sysroot=/opt/riscv/riscv32-unknown-elf --with-native-system-head
er-dir=/include --disable-libmudflap --disable-libssp --disable-libquadmath --di
sable-libgomp --disable-nls --disable-tm-clone-registry --src=../gcc --disable
-multilib --with-abi=ilp32 --with-arch=rv32imzicsr --with-tune=rocket --with-isa
-spec=20191213 --with-gmp=/opt/homebrew/opt/gmp --with-mpfr=/opt/homebrew/opt/mp
fr --with-mpc=/opt/homebrew/opt/libmpc 'CFLAGS_FOR_TARGET=-Os -mcmmodel=medlow
' 'CXXFLAGS_FOR_TARGET=-Os -mcmmodel=medlow'
Thread model: single
Supported LTO compression algorithms: zlib
gcc version 14.2.0 ()
```

Note: If you are using an Intel-based Mac or encounter issues, please contact the TAs for assistance.