# 实验步骤

## 环境配置

### 安装Linux操作系统

由于仿真需要使用到Verilator，因此需要准备一个Linux环境，可以使用虚拟机软件安装Ubuntu也可以使用Windows自带的WSL2安装对应的Ubuntu版本。

这里推荐使用22.04版本的Ubuntu。

### 安装Verilator

官方安装手册：[Installation — Verilator Devel 5.021 documentation](https://verilator.org/guide/latest/install.html)

在Linux终端中依次输入如下命令：

$sudo apt-get install git help2man perl python3 make autoconf g++ flex bison ccache

$sudo apt-get install libgoogle-perftools-dev numactl perl-doc

$sudo apt-get install libfl2 # 如果报错忽略即可

$sudo apt-get install libfl-dev # 如果报错忽略即可

$sudo apt-get install zlibc zlib1g zlib1g-dev # 如果报错忽略即可

$git clone https://github.com/verilator/verilator # 拉取verilator的仓库，执行过一次即可

# 每次进行安装时都需要输入:

$unset VERILATOR\_ROOT

$cd verilator

$git checkout v5.010 # 推荐安装使用5.010版本

$autoconf

$./configure

$make -j `nproc` # 如果报错请尝试直接输入 make命令

$sudo make install

经过以上步骤，verilator应该已经安装完毕，可以在终端中输入以下命令测试是否安装成功：

$verilator –version

如果正确的输出了版本号，则说明安装成功；如果报错，可以尝试重新安装。

### 安装gtkwave

在终端中输入，以下命令即可完成安装：

$sudo apt-get install gtkwave

### 安装java

先检查机器中是否已经安装有java环境，在终端中输入：

$java –-version

如果已经装有java，则可跳过此步骤。

$sudo apt-get install openjdk-11-jdk

$java –-version

输入上述命令后，如果成功输出版本信息，则表示安装成功。

### 安装mill

在终端中输入以下命令，即可完成安装：

$curl -L https://github.com/com-lihaoyi/mill/releases/download/0.11.6/0.11.6 > mill && chmod +x mill

$mv mill /usr/bin/mill

$mill --version

如果成功输出版本信息，则表明安装成功。

### 安装Vivado

### 获取实验框架

在不含中文字符的路径下解压实验框架的压缩包。

进入实验框架的目录中，输入以下命令：

$make verilog

如果在build目录下成功生成verilog文件，则说明上述所有的实验环境配置成功。

### 配置VS Code编程环境

[Download Visual Studio Code - Mac, Linux, Windows](https://code.visualstudio.com/Download)在官网里选择对应的deb版本安装包进行安装。

打开VS Code后在插件商店里搜索安装Metals和Scala Syntax这两个插件。这两个插件可以完成语法高亮、代码纠错、格式化等功能。

## 代码编写

在chisel/src文件夹下是CPU设计相关的chisel代码。根据不同实验的要求，在各chisel文件中进行代码的增减。完成代码设计后，在终端中输入make verilog命令后未发生报错，则说明代码没有基础错误，可以被综合实现；如果报错，请根据错误原因修改代码。

对于实验九而言，需要编写修改的内容在chisel/src/defines/isa/Instructions.scala和chisel/src/defines/isa/RVI.scala以及chisel/src/pipeline/decode/decoder.scala中。需要在decoder.scala文件中完成所有info信号的定义。在Instructions.scala中完成ALUOpType等object类的定义。在RVI.scala中填充新的指令的定义，以便在译码器中完成指令解码。

## 软件仿真

在输入make verilog后成功生成了verilog代码，这时再输入make func命令，观察命令行的输出，如果输出pass则表明通过，如果报错则在终端中输入make trace命令，这时会自动启动gtkwave软件打开刚刚CPU运行的波形图，请根据波形进行bug的修复。

## 硬件上板

在终端中输入make vivado，此时程序会自动调用vivado完成bit文件的输出，可以将bit文件发送至远程板中，观察数码管的显示结果是否与要求一致。若一致，则实现成功；否则请检查仿真是否通过。建议通过了仿真再进行上板，上板的耗时是仿真的几百倍。

## 提交评测

将生成的verilog代码粘贴至平台提交框中，提交代码即可完成评测。