DCS209 Lab1
实验报告

姓名 代骏泽
学号 24363012

目录

1 实验报告

1.1 环境配置

参照实验手册，本实验选择了 Windows 下的 Docker 环境方案.

1.2

2 后记

2.1 遇到困难

2.1.1 环境配置

其实在实验手册以及课程项目 repo 的帮助下，让整个环境 work 起来并不难，毕竟再不济也有 Docker. 但是我想讲的是我一开始尝试在原生 Windows (without WSL/Docker)下搭建环境的尝试.

2.1.1.1 java 相关

首先当然是安装构建系统以及配套所需的 jvm/jre

```powershell
scoop bucket add java
scoop install sbt
scoop install graalvm19-jdk11
```

> **Info**
>
> 注意 java 版本的选择，测试 jvm21 是无法兼容作业配的环境的，而 jvm19 是可以的
>
> 当然你还可以 `scoop install jenv` 管理 jvm 环境

2.1.1.2 verilator 相关

现在尝试运行 test 其实已经大差不差了，至少 java 相关的环境不会报错：

```powershell
cd lab1
sbt test
```

但是会报错找不到 `verilator_bin`：

```
java.io.IOException: Cannot run program "verilator_bin" (in directory "C:
\Users\DvdBr3o\devenv\dcs209\lab1"): CreateProcess error=2
```

所以现在只差一个 `verilator` 我们就完事大吉了！但巧也不巧的是，不同于 `iverilog`，`verilator` 恰好不支持原生 Windows build。你在 scoop 上找不到它的 distribution，在它 github repo release 里也找不到 Windows build. ~~这真是太棒了!~~

而当你翻看 `verilator` 的 github repo 时，你会惊讶的发现它竟然是个 all in C++ with CMakeLists.txt… 而当你尝试 clone 下来 build 时发现它还不是那种 “another Makefile” 的 CMake project，它在 Windows 下好像真的可以 build 出来. 事实上，它 repo 的 `./ci` 下还有个 `./ci/ci-win-compile.ps1`. ~~留个 build script 但不给官方 build release? 有点意思.~~

2.1.2 三个 CPUTest

2.2 课程建议

有点像无病呻吟所以不分小标题了，大概就几句小牢骚：

1. 工具选择. 我不知道 Chisel 在硬件领域这块的权威性，也不知道是不是我们院老师们都有点 jvm 情节. Chisel 确实比 verilog 语义上开起来更现代更优雅，工具链也相比没那么古老了. 但是它环境真的很难配，java 环境很重，用在 non serving 的硬件验证 & 综合这块感觉过于笨重了. 而且最后其实基本都靠 docker 一把糊过去 (而实验手册 [pur25] 中给出的 devcontainer 方案，实则是跑在挂 volume 的 docker 下，而挂 volume 而不是直接在环境自己的文件系统里的话，file io 会慢很多，导致 build & test 处处变慢，开发体验会很差，救救买不起第二个笔记本装 linux 的学生吧). 或许可以选择更好配环境的工具链替代 Chisel.
2.

引用