



计算机系统贯通课程-计算机系统 I

—— Lec0x2: 一些注意事项& 流程说明

洪奕迅 3230102930@zju.edu.cn

史璐欣 3220104390@zju.edu.cn

计算机学院
浙江大学

2025 年 2 月 27 日



实验相关

- 出于大家的熟悉程度考虑，Lab1 整体往后延迟一周，从下周开始持续三周。
- 很多同学反映的实验课和理论课天差地别：因为实验和理论课需要包含的内容不完全相同，对于进制浮点数以及四则运算等的内容在理论课是不得不提的，相应的理论课留给数字逻辑的内容并不会有实验课那么多，对于基本器件也讲得更快，而这些器件在实验课需要留出充足的时间给大家设计代码，这种情况也不会持续太久，因此希望同学们理解，对于实验文档中任何难以理解的地方都可以私信我们说明，我们会补充文档。
- 实验报告：**务必上传 pdf 避免格式崩坏**，对于样式没有太多要求，但写之前记得查看实验文档注意事项，至少保证阅读清晰，顺序合理。
- 问问题：多发多看 issue，私聊问问题尽量用钉钉，这样可以减少大家询问 TA 和 TA 重复劳动的时间。

确定工作区

- 在开始实验之前大家至少知道自己在做什么实验，对于我们目前的实验而言，需要大家修改的代码文件通常在 `sys1-sp25/src/labxx` 下，而我们提供的代码文件通常在 `sys1-sp25/repo/labxx` 下
- 大家大多使用的是 Linux 的命令行，因此在执行各种指令时请务必搞清楚自己在什么路径以及需要在什么路径执行，最好了解一下绝对路径和相对路径概念，可以有效避免很多执行文档指令路径不对造成的错误（例如 `git submodule update --init repo/sys-project`）。

```
forever@forever:~$ pwd  
/home/forever
```

```
forever@forever:~$ cd sys1-sp25  
forever@forever:~/sys1-sp25$
```

```
forever@forever:~$ cd /home/forever/sys1-sp25  
forever@forever:~/sys1-sp25$
```



你需要会的

需要的工具及了解程度

- Linux 终端: 因为我们通过 Linux 终端与 WSL/虚拟机交互, 你至少要会在正确位置复制粘贴我们提供的指令, 你还要会看一些基本的错误 (或者至少会搜索报错信息).
- Git: 如果你不需要维护自己 fork 的仓库, 那么除了文档提供的指令外你只需要经常 `git pull` 拉取我们的更新即可.
- Verilog: 你可以使用任何编辑器, VSCode/Vivado 甚至记事本写 verilog, 但写代码的时候请务必在脑中想象电路, 至于语法丰俭由人, Lab0-2 只需要你会 `module`, `assign`, `wire` 即可
- Verilator: 在大家执行 `make/make verilator` 时其实都是在使用 Verilator 进行 Verilog 代码的编译, 因此大家要学会和看 C 语言报错一样会看 Verilog 的报错
- Makefile: 每次执行 `make` 指令, 系统会调用 **当前目录下的 Makefile**, 可以查看 Makefile 来了解我们提供的自动化指令具体做的事. 此外, 也要学会看 Makefile 的报错.
- GTKWave: 用来查看仿真的波形, 主要用来 Debug, 但目前实验的复杂程度大家暂时只需要会把端口拖进去看波形变化即可.

通常写 Verilog 的工作流程

- 明确电路/逻辑表达式
- 按照搭建电路的顺序开始写代码
- 设计仿真文件：要覆盖重要的情况，但并不一定要完全遍历。
- 编译查看是否有误：学会看报错，Verilator 事实上是一个把 Verilog 转为 C++ 的编译器，而在终端输出的就是编译信息，这里面包含的报错信息通常是可读的，它的含义一般就对应你语法上或者电气上的错误，debug 的过程本身也是你学习语法和熟悉电路的过程。

举个 🍌：

$$F(A, B, C, D) = ACD + \overline{A}BC\overline{D}$$



上板流程

- 编写引脚约束文件 (仓库中的 `syn/nexysa7.xdc`)
- 编写一个顶层文件 (`top.v`), 对于我们这种只有一个模块的 Lab, 这个看起来有点多余, 但是后面的 Lab 会体现其作用.
- Vivado 生成比特流后上板测试 (`make bitstream`/手动 Vivado 操作)