

实验测试方法

首先，clone 代码仓库：

```
1 cd 2022Spring/ # 选择一个合适的目录
2 git clone git://github.com/FDUCSLG/Arch-2022Spring-FDU.git
3 cd Arch-2022Spring-FDU
4 git submodule update --init --recursive
5 make test-lab1
```

代码编写

我们在 `vsrc` 里写 SystemVerilog 代码。

有以下要求：

- 每个 `.sv` 文件仅编写一个 module
- 文件名与 module 名一致，一般为全小写
- 每个文件都要用 ``ifndef __NAME_SV` 来保护
- 每个文件都要有：

```
1 `ifdef VERILATOR
2
3 `else
4
5 `endif
```

- verilator 仿真中，module 进行元件例化前，需要以 `vsrc/` 为根的相对路径 include 源文件，写在 ``ifdef VERILATOR` 区域。
- vivado 中，只需 include 头文件，include 时不写路径，只写 文件名 `.svh`。
- module 的一般形式：

```
1 module sample
2     import common::*; #(
3     parameter XLEN = 16
4 ) (
5     input logic clk, reset
6 );
7
8 endmodule
```

Verilator 仿真

这是我们主要的调试手段。

运行 `make test-lab1` 等测试，观察输出。如果有 `Hit Good Trap` 输出则为通过。

Vivado 仿真

请在 Verilator 仿真通过后再尝试 Vivado 仿真！

打开 `vivado/test1/project/project_1.xpr`，运行 `vsrc/add_sources.tcl` 添加源文件，然后 `Run Simulation`。点击上方三角开始仿真。

Vivado 上板

请在 Verilator 仿真通过后再尝试上板！

点击 `Generate Bitstream`，生成完毕后连接实验板并 `Program Device`。

在电脑上打开串口软件（Windows 系统可使用 elearning 上发的 SecureCRT，其他系统请自行寻找相关软件），设置串口（Serial）波特率（Baud Rate）9600，然后连接。

按下实验板上的 `prog`，可在串口软件中看到一行输出。

如果你通过了第一个实验的测试，串口软件上会输出 `Hello world!`。

以后的实验都需要从串口中观察输出。