

计算机组织与体系结构实习Lab 2.1

2025年春

1、RISC-V交叉编译工具链

下载并安装RISC-V交叉编译工具链，参考群里的“RISC-V实验环境说明”。

注意：本lab需要使用的是RISC-V的64位交叉工具链riscv64-unknown-elf-gcc。

2、RISC-V指令级模拟器

首先下载实验代码框架：

<https://disk.pku.edu.cn/link/AAA73B4DCE3A8F43DC893501BD3AE56D12>

文件名：simulator.tar.gz

有效期限：2025-07-01 14:45

解压后进入simulator目录，终端输入：

```
make T=dummy
```

最后几行输出：

```
-----Start Simulation-----
[INFO] (src/memory.c:54) Physical Memory Range:[0000000080000000,
0000000087ffffff].
[INFO] (src/memory.c:60) The image is test/build/dummy.bin, size = 57.
HIT GOOD TRAP!
[INFO] (src/cpu.c:35) Program ended at pc 80000030, with exit code 0.
```

则说明构建成功。

3、实验框架说明

1. `sim` 目录：存放RISC-V指令级模拟器实现代码。
2. `test` 目录：存放测试程序代码，共有10个测试程序。
3. `trace` 目录：存放用于调试的PC踪迹。
4. `make T=dummy`：表示以 `dummy.c` 为测试程序，运行RISC-V指令级模拟器。

4、检查要求

==基本要求==：

1. 仔细阅读框架代码，理解整个项目的构建流程以及模拟器是如何执行的，将你的理解写在实验报告中。
2. 为模拟器设置两种运行模式，Batch模式下直接执行至结束，Debug模式下能够实现以下功能。

命令	格式	使用举例	说明
帮助	<code>help</code>	<code>help</code>	打印命令的帮助信息

命令	格式	使用举例	说明
继续执行	c	c	继续执行被暂停的程序
退出	q	q	退出模拟器
单步执行	si [N]	si 10	让程序单步执行N条指令后暂停执行，当N没有给出时，缺省为1
打印当前执行状态	info r	info r	打印寄存器状态（包括PC）
扫描内存	x N ADDR	x 10 0x80000000	将ADDR作为起始内存地址，以十六进制形式输出连续的N个4字节

3. 实现**部分RV64IM**指令，测试通过(HIT GOOD TRAP)全部10个测试用例，并提供可以批量测试全部用例的命令。
4. 最终提交：实验报告（格式不限），项目源码，执行脚本及说明。

==加分项==：

1. 实现**全部RV64IM**指令，测试通过助教私有（不对外公布）的另外5个测试用例。
2. 实现**系统调用**功能。

提示：

你除了需要实现系统调用相关的指令之外，还需要添加一些特殊的系统寄存器，叫做控制状态寄存器（Control and Status Register）。目前，你可以只实现下面4个CSR。

- o mtvec寄存器：存放异常入口地址
- o mepc寄存器：存放触发异常的PC
- o mstatus寄存器：存放处理器的状态
- o mcause寄存器：存放触发异常的原因

触发异常后的硬件响应过程如下：

1. 将当前PC值保存到mepc寄存器
2. 在mcause寄存器中设置异常号
3. 从mtvec寄存器中取出异常入口地址
4. 跳转到异常入口地址

注意：上述保存程序状态以及跳转到异常入口地址的工作，都是硬件自动完成的，不需要程序员编写指令来完成相应的内容。

实现系统调用后，理论上，你就能实现所有操作系统的功能。例如**输入输出、进程切换、虚拟内存和文件系统**等。请自行编写测试用例测试你所实现的各项功能。

3. 实现RV64的其他扩展集，并自行编写测试用例测试。
4. 为模拟器实现各种trace功能（可随时开关）。

trace功能	说明	提示
itrace	指令踪迹，应当包含PC，指令十六进制表示，指令的反汇编	可以链接llvm库实现指令的反汇编
mtrace	访存踪迹，应当包含访存地址，读/写，访存数据；不包括取指过程	\
ftrace	函数调用踪迹，输出参考格式见附录	解析elf文件，查找符号表位置得到函数名

5、附录

ftrace输出参考格式 (dummy)：

```
0x8000000c: call [_trm_init@0x80000018]
0x80000028:  call [main@0x80000010]
0x80000014:  ret  [main]
```