2021 ESWIN 软件技术培训

RISC-V Lab

RISC-V Lab的获取

- RISC-V Lab相关的代码、说明文档、测试样例都打包在riscvlab.tar中，将会发送到答疑群里。

建议的环境

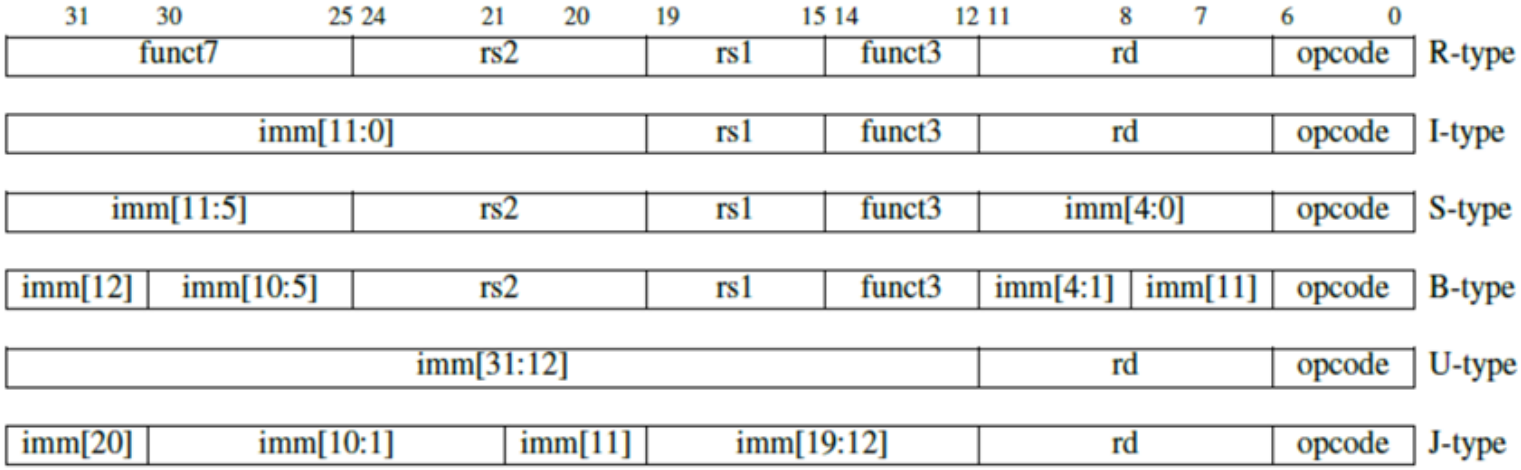
- **实验环境：Linux 系统，如 Ubuntu**
- **如果windows环境下可以使用 virtualbox 来运行 ubuntu 等虚拟机，提供一个参考的安装教程**
- **提示：**
  - 要求在ubuntu下载好 gcc 和 make 用于编译
  - 可以在ubuntu中自由使用编辑器或IDE，推荐使用vscode
  - 要求学会使用linux的命令行工具terminal

RISC-V

- RISC-V是一个最近诞生的指令集架构（它诞生于近十年，而大多数其他指令集都诞生与20世纪70-80年代），也是一个开源的指令集架构。
- 现代x86和ARM架构采用了增量ISA，为了保持架构的向后兼容性使得架构的篇幅已有数千页。而新兴的RISC-V架构则具有后发优势，采用模块化的ISA，其核心是名为RV32I的基础ISA，可以在此基础上根据不同的应用场景扩展其他的模块，实现不同的要求。

RV32I基础指令集

- RV32I是RISC-V固定不变的基础整数指令集，是RISC-V的核心内容。
- RV32I指令有六种格式，分别是R型、I型、S型、B型、U型、J型指令。每条指令均为32位长。



RV32I基础指令集

- RV32I每条指令的具体布局、操作码等均可以在Writeup或RISC-V-Reader-Chinese-v2p1.pdf中找到。
- RISC-V-Reader-Chinese-v2p1.pdf的第二章讲的就是RV32I指令集，大家可以阅读一下。

31 25	24 20	19 15	14 12	11 7	6 0	
imm[31:12]				rd	0110111	U lui
imm[31:12]				rd	0010111	U auipc
imm[20|10:1|11|19:12]				rd	1101111	J jal
imm[11:0]		rs1	000	rd	1100111	I jalr
imm[12|10:5]	rs2	rs1	000	imm[4:1|11]	1100011	B beq
imm[12|10:5]	rs2	rs1	001	imm[4:1|11]	1100011	B bne
imm[12|10:5]	rs2	rs1	100	imm[4:1|11]	1100011	B blt
imm[12|10:5]	rs2	rs1	101	imm[4:1|11]	1100011	B bge
imm[12|10:5]	rs2	rs1	110	imm[4:1|11]	1100011	B bltu
imm[12|10:5]	rs2	rs1	111	imm[4:1|11]	1100011	B bgeu

顺序模拟器

- 模拟机器代码的执行。
- 实现高效的、流水线化的模拟器的第一步。
- 处理的各个阶段（书本P264）

  取指、译码、执行、访存、写回、更新pc
- 取指：icode、ifun、rA、rB、valc、valp
- 译码：vala，valb
- 执行：vale
- 访存：valm，内存地址
- 写回：寄存器
- 更新PC：下一条指令地址

Risc-v lab介绍

- 本次实验你将设计并实现一个risc-v的顺序模拟器，你将通过扩展该模拟器以支持更多指令的方式来了解该模拟器。
- 该实验分为3个部分，内容均是实现不同的指令，但是难度会一点点的提升，重在理解一条指令在模拟器中各个阶段应该进行的操作。
- 需要实现的指令看似很多，但是大多只是功能位不同，每个部分实现了一个其他的也就迎刃而解，不要被数量吓倒。

Datalab自测评分

- 每当你修改后便可在命令行输入
- unix> gcc -Wall -O2 -o ssim hcl.c ssim-simple.c isa.c
- 获得可执行文件，再输入
- unix>./ssim -t *.yo（*.yo为后缀为.yo的文件）
- 获得运行结果。
- 与Writeup的附录中的输出结果进行对比，以求证是否正确。

RISC-V Lab注意事项

- **需要仔细阅读的内容包括：**

  isa.c文件中的instruction_set：

  isa.h文件中的itype_t：

  hcl.c文件全部

  ssim-simple文件中的sim_step、 update_state函数

- **需要修改的位置**

```
////////////////////////////////////
//PART A:
////////////////////////////////////
```

- **ifun1即图中的funct3， ifun2即图中funct7， valc中存的即imm立即数。**

RISC-V Lab注意事项

- imm立即数获取之后再向左平移再向又平移是因为立即数是符号扩展的，需要将获取的32位的最高位变为符号位。
- 注意Part B部分在获取imm立即数时会有一点不同（位移的立即数版本不需要有符号）
- 根据注释理解函数的含义
- 举例：以I_R为例

附加题

- 如果对于risc-v指令比较熟悉的同学对于上述实验做起来太快，可以更进一步。
- 在将顺序模拟器改造成流水线模式之前，我们需要进行简单的调整以适合流水线模式。
- 将更新PC阶段放在一个周期开始时，而不是结束时。具体的结构参照SEQ+（P288）。
- 这部分内容不会出题，请有余力和兴趣的同学进行尝试。（也不难的）

2021 ESWIN 软件技术培训

Q & A

2021.8