**深 圳 大 学 实 验 报 告**

**课 程 名 称： 计算机系统(3)**

**实验项目名称： 取指和指令译码设计**

**学 院： 计算机与软件学院**

**专 业： 计算机与软件学院所有专业**

**指 导 教 师： 罗秋明**

**报告人： 岳海涛 学号： 2022152020 班级： 22计科**

**实 验 时 间： 2024年10月18日**

**实验报告提交时间： 2024年10月22日**

**教务处制**

**一、 实验目标：**

设计完成一个连续取指令并进行指令译码的电路，从而掌握设计简单数据通路的基本方法。

**二、实验内容**

本实验分成三周（三次）完成：1）首先完成一个译码器（30分）；2）接着实现一个寄存器文件（30分）；3）最后添加指令存储器和地址部件等将这些部件组合成一个数据通路原型（40分）。

**三、实验环境**

硬件：桌面PC

软件：Linux Chisel开发环境

**四、****实验步骤及说明**

本次试验分为三个部分：

1. 设计译码电路，输入位32bit的一个机器字，按照课本MIPS 指令格式，完成add、sub、lw、sw指令译码，其他指令一律译码成nop指令。输入信号名为Instr\_word，对上述四条指令义译码输出信号名为add\_op、sub\_op、lw\_op和sw\_op，其余指令一律译码为nop，输出信号均为1bit。

给出Chisel设计代码和仿真测试波形，观察输入Instr\_word为add R1,R2,R3; sub R0,R5,R6，lw R5,100(R2), sw R5,104(R2)、JAL 100时，对应的输出波形。

Instr\_decoder

Instr\_word[31:0]

add\_op

sub\_op

lw\_op

sw\_op

nop

1. 设计寄存器文件，共32个32bit寄存器，允许两读一写，且0号寄存器固定读出位0。五个输入信号为RS1、RS2、WB\_data、Reg\_WB、RF\_WrEn，寄存器输出RS1\_out和RS2\_out；寄存器内部保存的初始数值等同于寄存器编号。

给出Chisel设计代码和仿真测试波形，观察RS1=5，RS2=8，WB\_data=0x1234，Reg\_WB=1，RF\_WrEn=1的输出波形和受影响寄存器的值。

Reg\_file

RS1[4:0]

RS2[4:0]

WB\_data[31:0]

Reg\_WB[4:0]

RS1\_out[31:0]

RS2\_out[31:0]

RF\_WrEn

1. 实现一个32个字的指令存储器，从0地址分别存储4条指令add R1,R2,R3; sub R0,R5,R6，lw R5,100(R2), sw R5,104(R2)。然后组合指令存储器、寄存器文件、译码电路，并结合PC更新电路（PC初值为0），最终让电路能逐条指令取出、译码（不需要完成指令执行）。

给出Chisel设计代码和仿真测试波形，观察四条指令的执行过程波形，记录并解释其含义。

PC

（初值为0）

+4

指令内存

Instr\_decoder

Reg\_file

RS1[4:0]

RS2[4:0]

WB\_data[31:0]

RS1\_out[31:0]

RS2\_out[31:0]

add\_op

sub\_op

lw\_op

sw\_op

nop

Reg\_WB[4:0]

RF\_WrEn

**五、实验结果**

**1.环境配置**

首先我们对实验的环境进行配置：

下载老师提供的虚拟机文件：

大概下了4个小时左右，将其导入到虚拟机vmware station报错



解决方法：

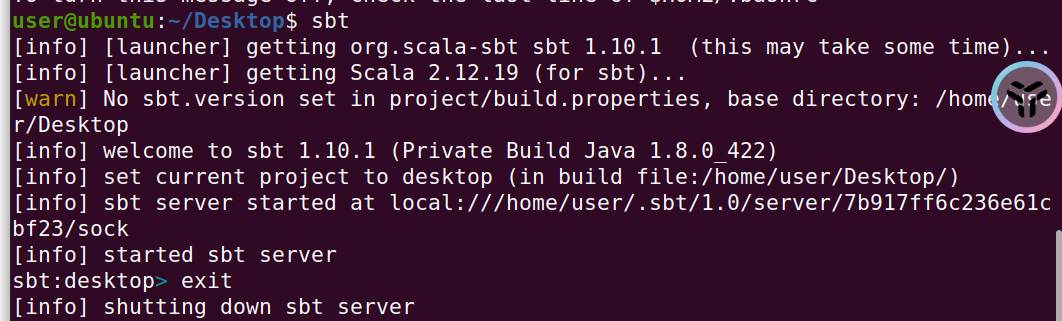
1.找到虚拟机的存放文件夹路径，并通过记事本程序打开虚拟机的.vmx文件；

2.找到文件中的virtualHW.version = "20"字段，修改该字段为virtualHW.version = "16"

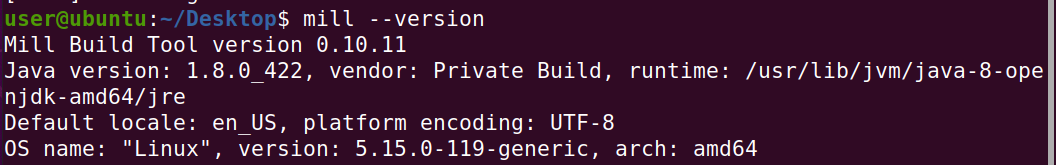
3.保存.vmx文件，并开启虚拟机。

开机对实验环境进行验证：

sbt



mill

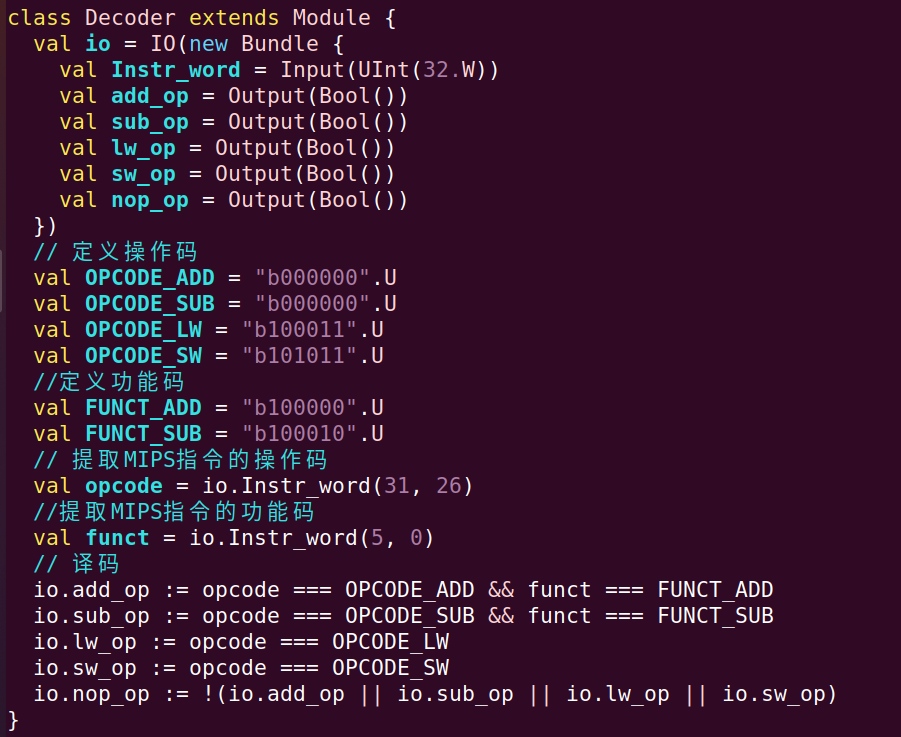


verilator

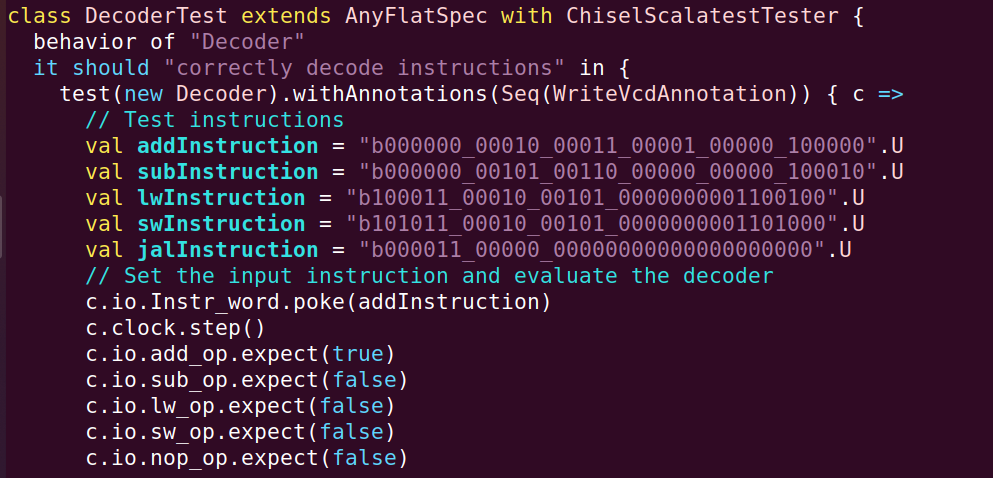


大功告成，开始实验：

1. **设计译码电路**

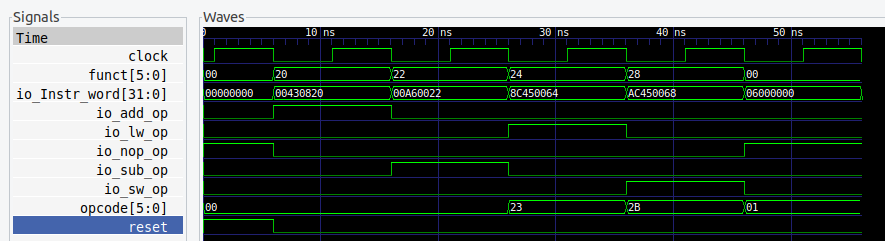


**测试：**

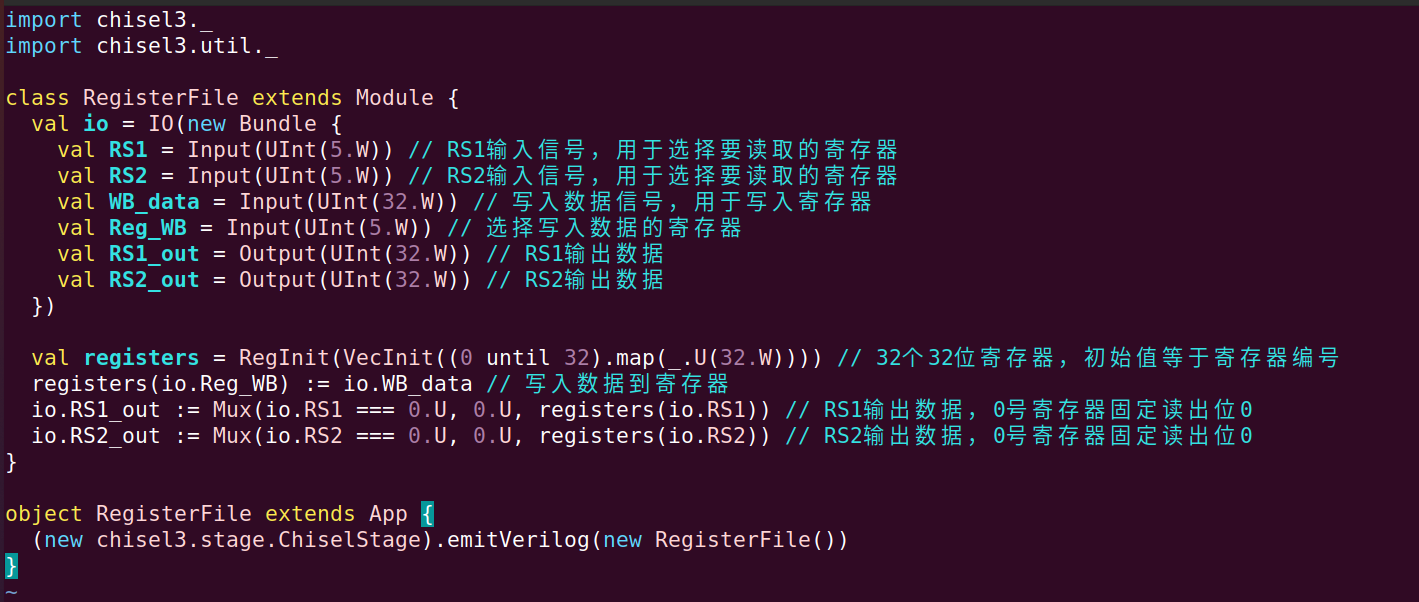


**测试其他指令大差不差。**

**有图有真相**



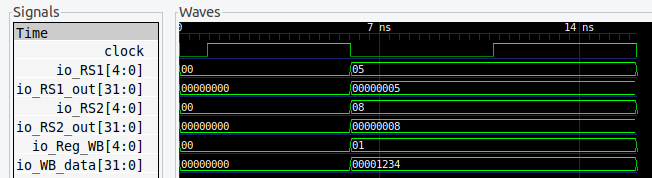
**设计寄存器文件：**



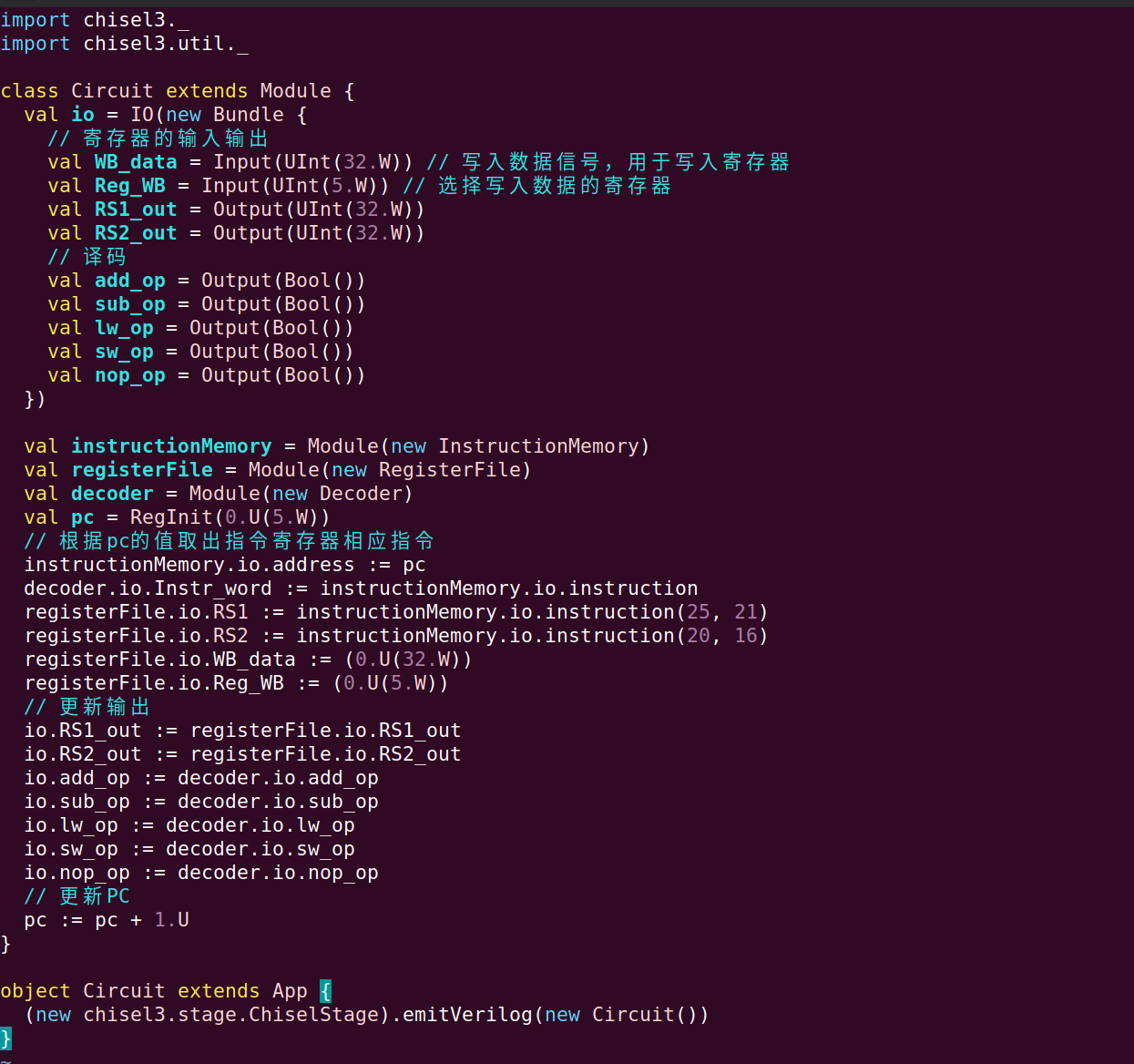
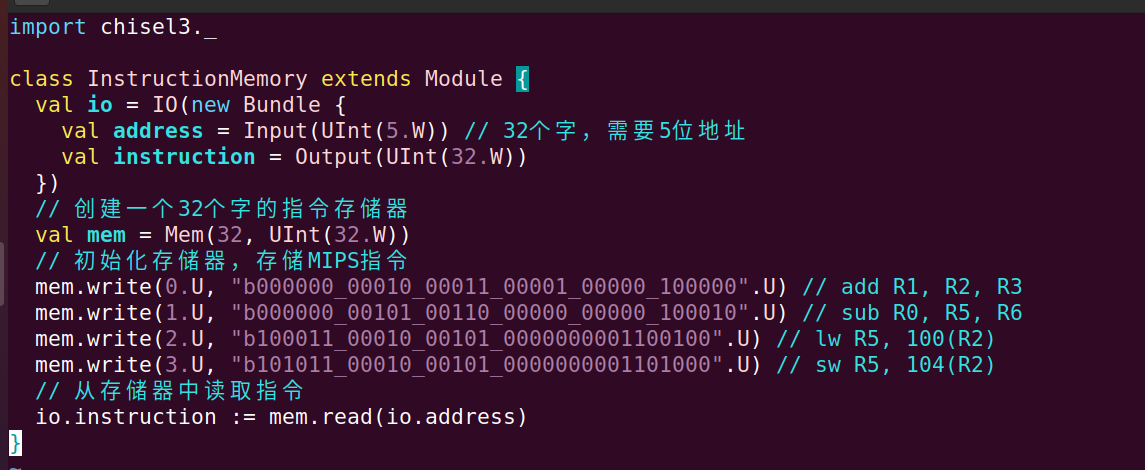
**测试文件：**



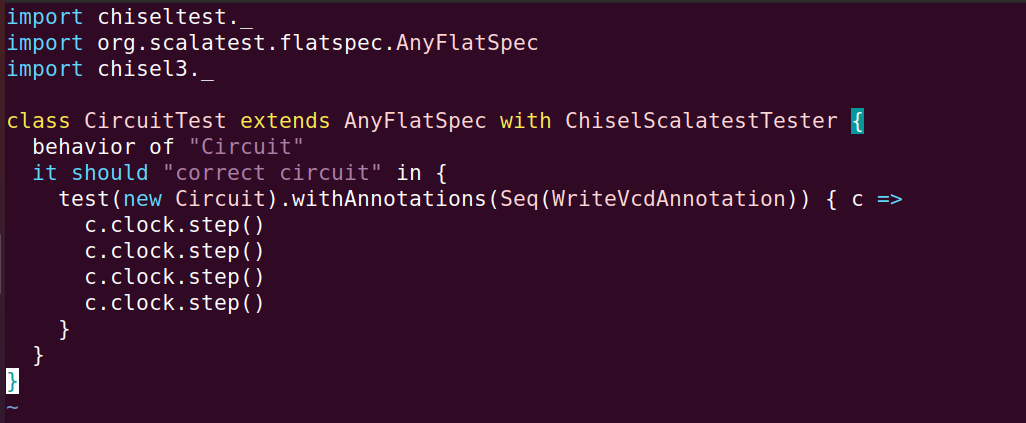
**测试波形：**



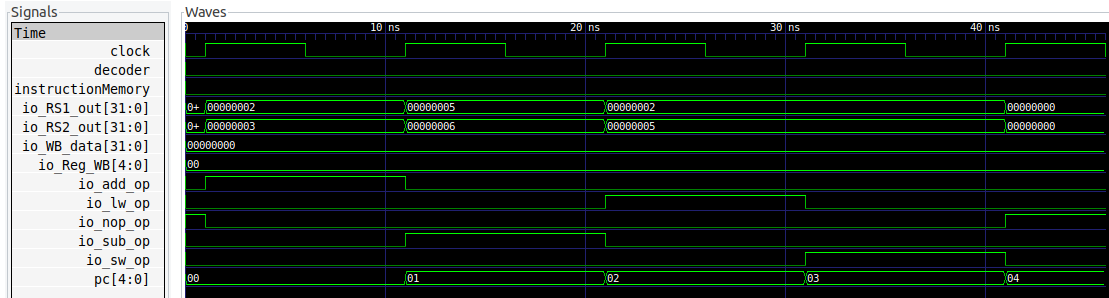
**实现一个32个字的指令存储器**



**测试：**



**测试波形：**

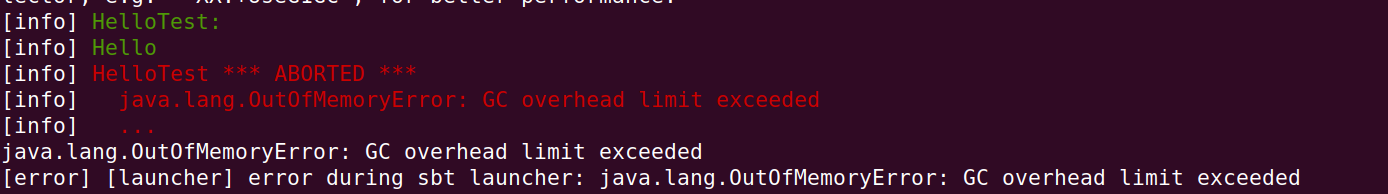


**五、实验总结与体会**

**在这里要记录一些莫名奇妙的坑，我是跟着老师的视频在github上拉取了项目之后一步一步做。**

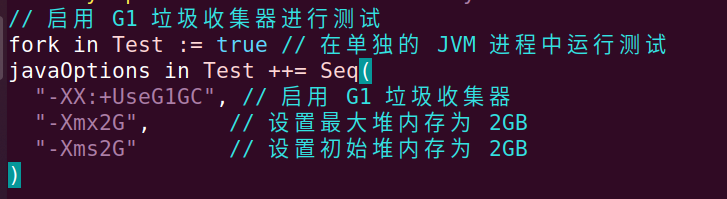
1. **git clone失败，没办法，多来几次，终于成功**
2. **sbt run失败，一开始是网络问题，下载依赖失败，多来几次就可以了**

**后面jvm报错：**



**显示heap内存不够，gc裂开了**

**解决方法：调大堆内存**



|  |
| --- |
| **指导教师批阅意见：**  **成绩评定：**  指导教师签字：  年 月 日 |
| 备注： |