# 实验零—单周期CPU设计

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 姓名： | 汤尧 | | 学号： | 3200106252 | | | 学院： | 求是学院云峰学园 | | |
| 课程名称： | | 计算机系统Ⅱ | | | 同组学生姓名： | 无 | | | |
| 实验时间： | | 周四3，4，5节课 | 实验地点： | | 紫金港机房 | | 指导老师： | | 卢立 | |

# 实验目的和要求

1. 回顾单周期CPU数据通路和控制器设计过程
2. 回顾程序执行过程
3. 为之后搭建流水线CPU打下基础

# 实验内容和原理

## 实验内容

1. 完成32位单周期CPU数据通路设计
2. 完成单周期CPU控制模块设计，结合实验10数据通路，搭起32位单周期CPU
3. 实现指令
4. 通过仿真测试和上板验收

## 设计模块

2.2.1概述

单周期 CPU 主要特征是在一个周期内完成一条指令，也就意味着其 CPI(cycle per instruction) 是 1。考虑到不同的指令都需要在一个周期内完成，因而单周期 CPU 时钟频率是以执行最慢的那条指令作为基准，这也是单周期 CPU 最大的缺陷之一。

我们可以把单周期 CPU 分成数据通路和控制单元两个模块，本次实验将完成数据通路模块。

2.2.2 数据通路设计

为了之后能和流水线 CPU 进行衔接，我们把单周期 CPU 数据通路划分成 5 个阶段(stage):

1. instruction fetch (IF)

2. instruction decode (ID)

3. execution (EX)

4. memory access (MEM)

5. register write back (WB)

• 获取指令阶段 (IF)

– 从 instruction memory 中获取 32 位指令

– 令当前的 PC(program counter) 增加 4（PC = PC + 4，使当前 PC 指向下一条 32

位的指令，instruction memory 是 byte-addressing，所以是 +4 而不是 +32）

• 指令译码阶段 (ID)

– 拿到指令之后，我们需要知道指令能提供各种信息包括 opcode, rs1, rs2, rd, imm

等等，所以需要对指令进行译码，提取信息。

– 译码完成后，我们可以通过拿到的 rs1, rs2 读取相应的寄存器值

• 执行阶段 (EX)

– 执行一些运算操作，包括 +，-，\*，/，逻辑运算或左移、右移等等

– 跳转地址的计算也可以在这个阶段执行

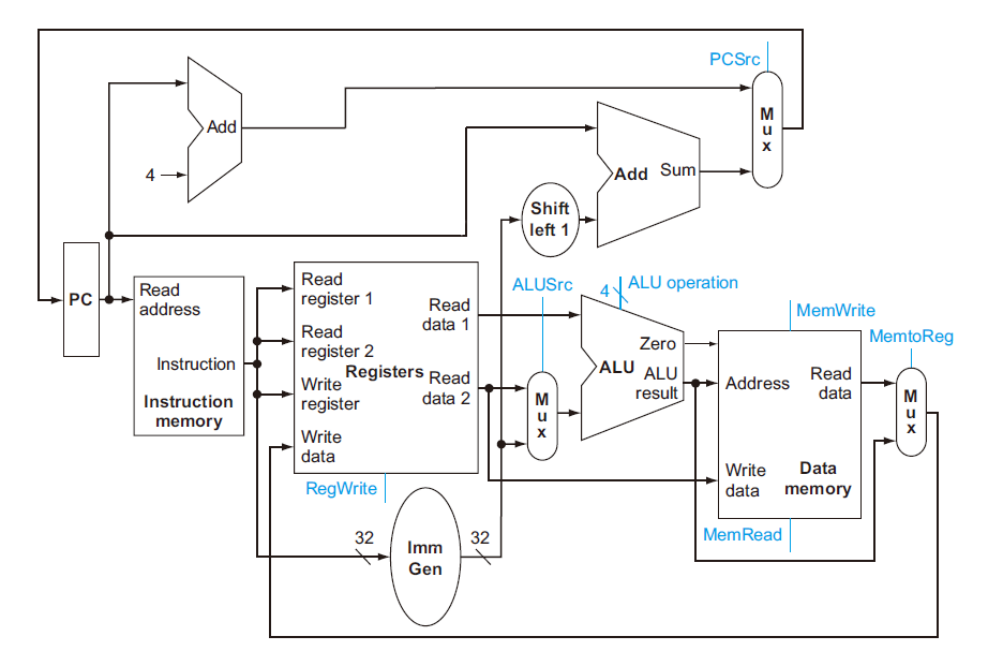
• 访存阶段 (MEM)

– 从 data memory 中拿到数据

• 写回阶段 (WB)

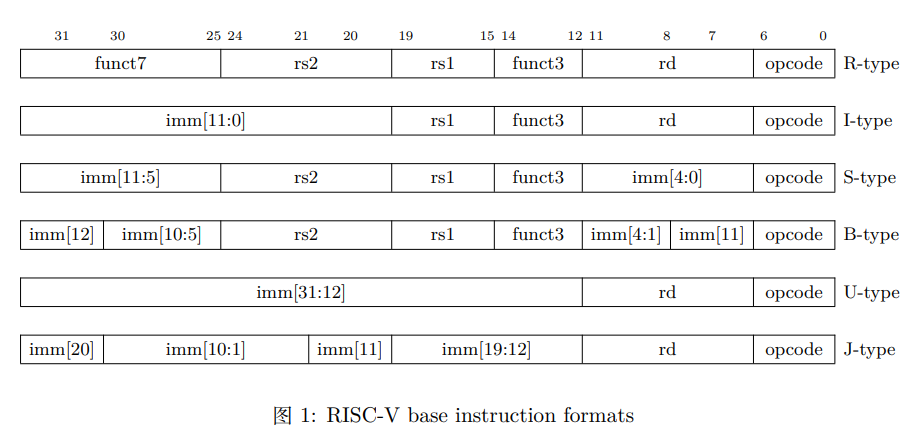
– 把计算后获得的数据或者从 data memory load 的数据写回到寄存器中

2.2.3 数据通路图

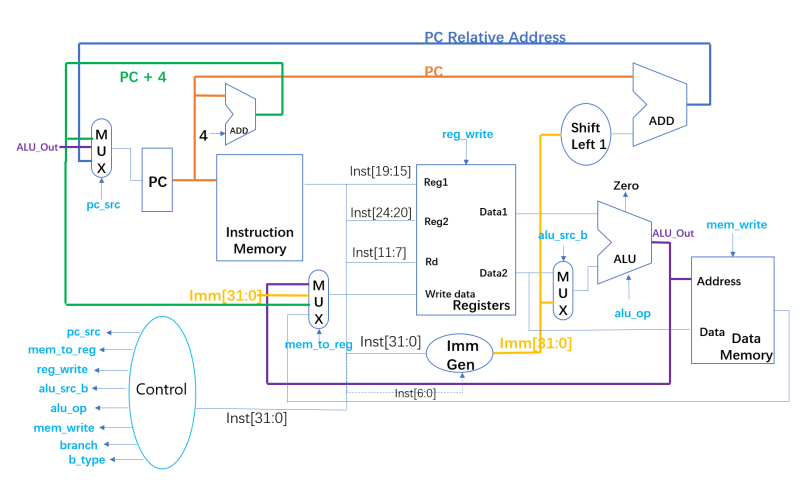


2.2.4 控制单元

控制单元，也称译码器，它的作用是解码指令，发出信号，告诉 Datapath 应该执行什么操作。



2.2.5 单周期CPU参考设计



2.2.6 实现指令

I型： addi,slli,slti,sltiu,xori,srli,ori,andi,srai,lw,jalr

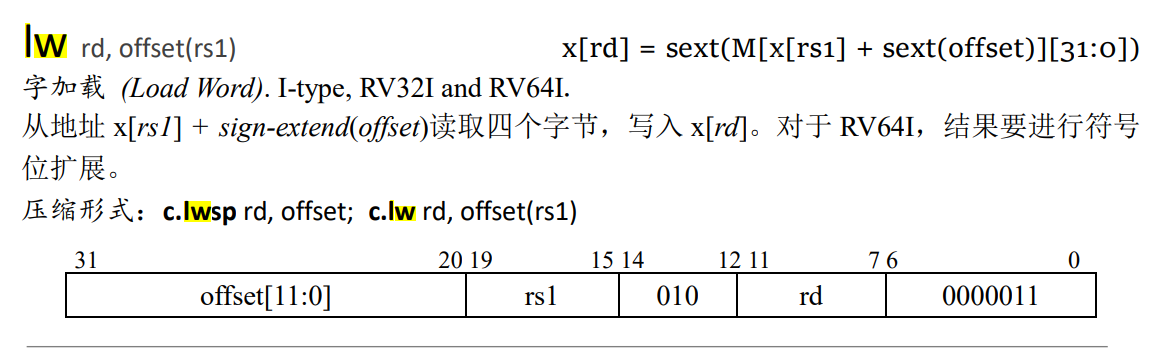
R型：add, sll, slt, sltu, xor, srl, or, and, sra

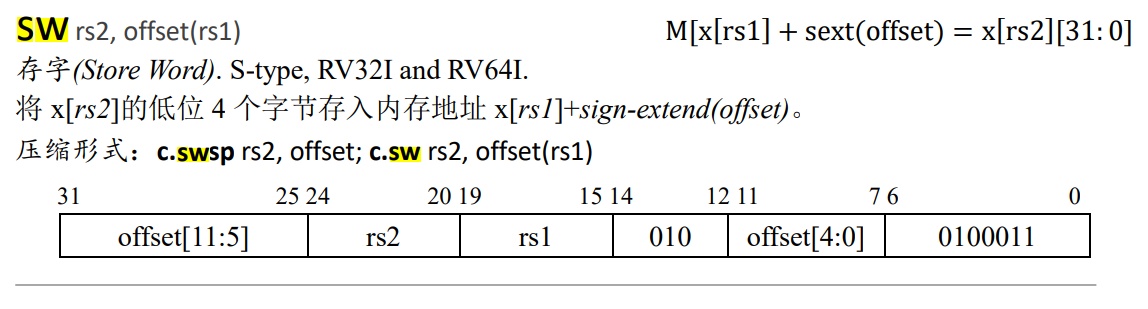
U型：lui, auipc

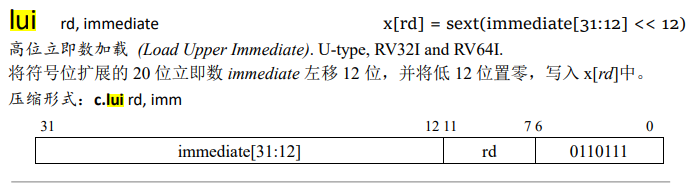
S型：sw

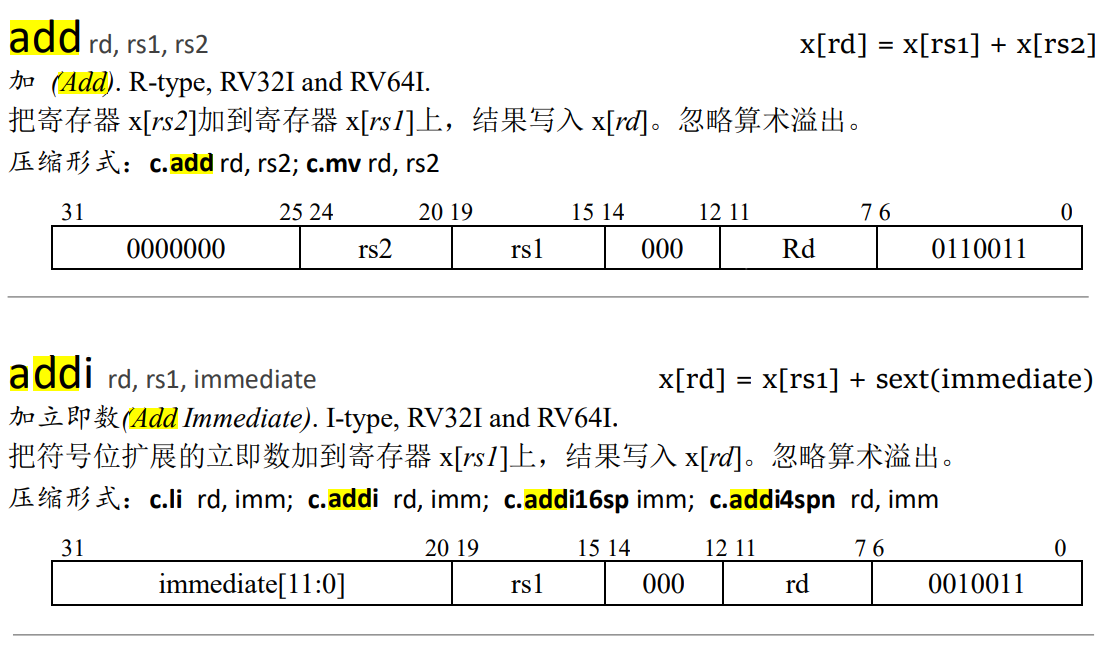
B型：beq, bne

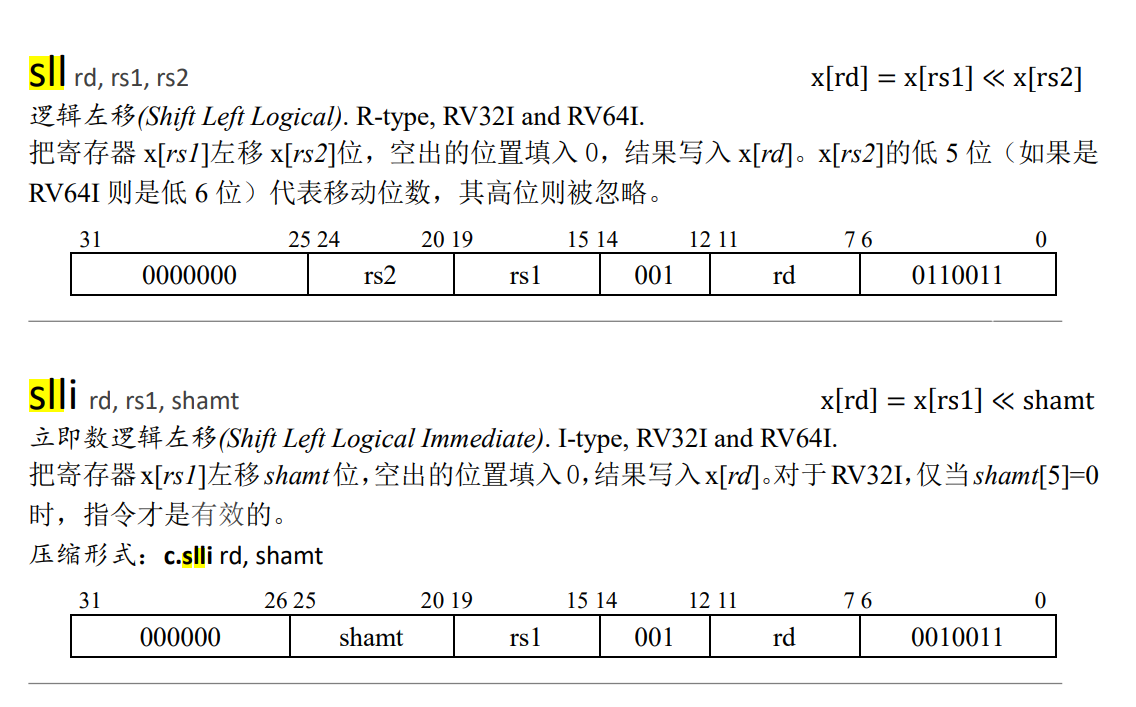
J型：jal

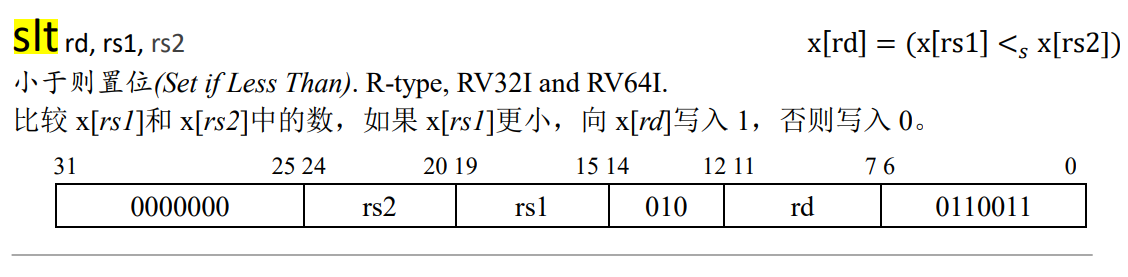


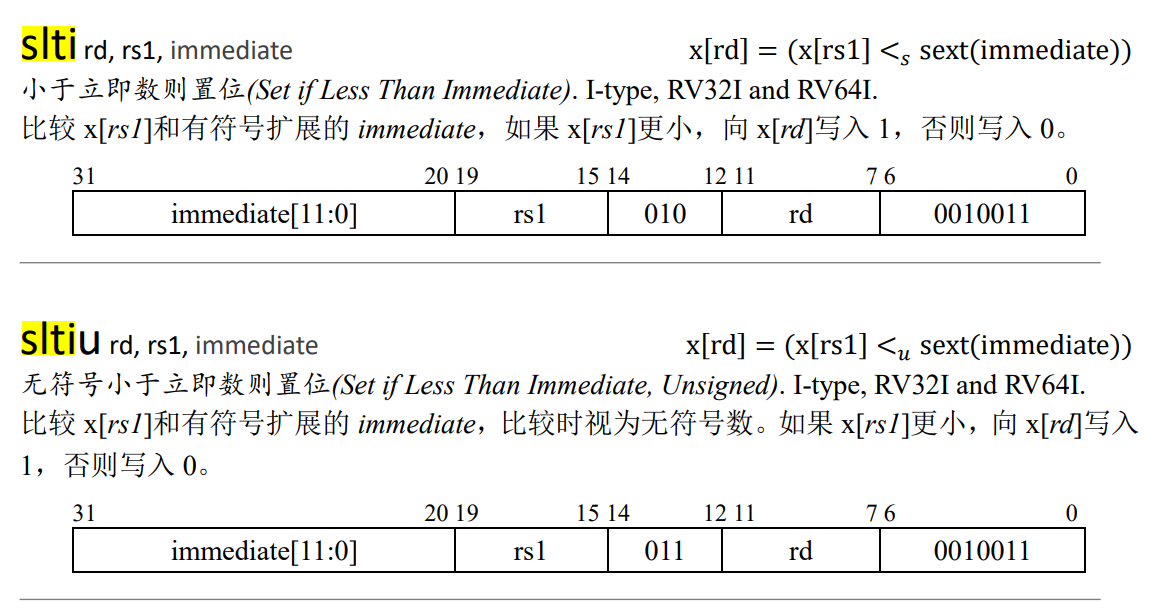


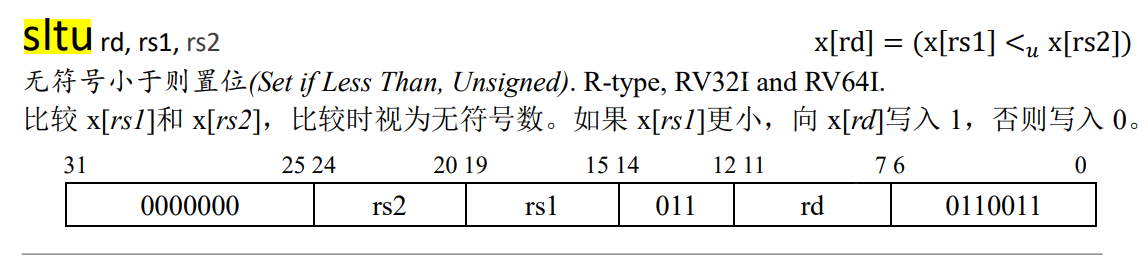


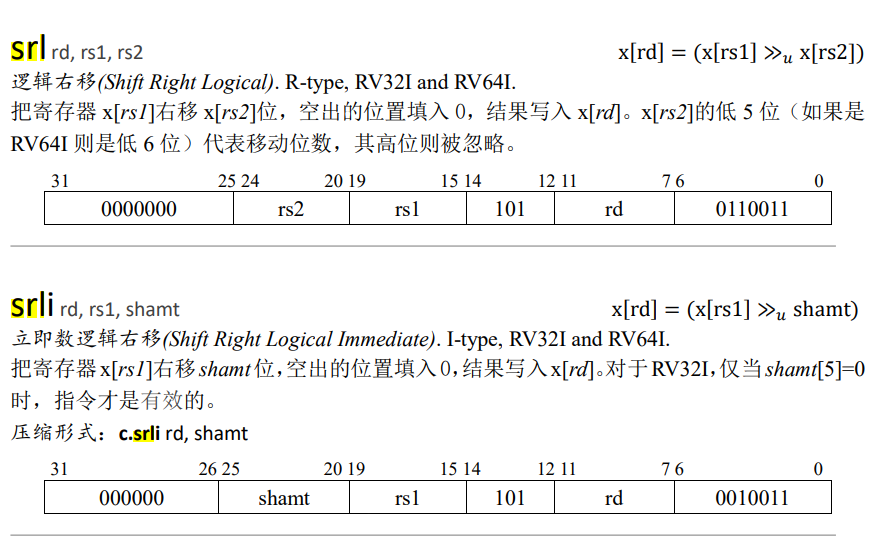


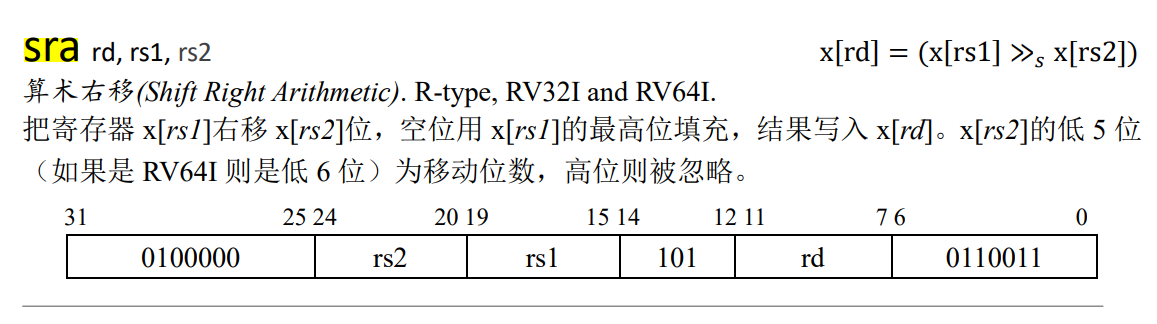


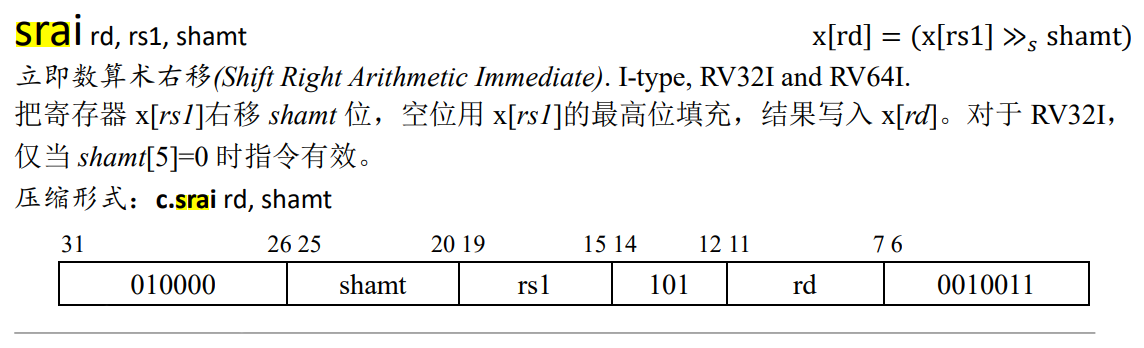


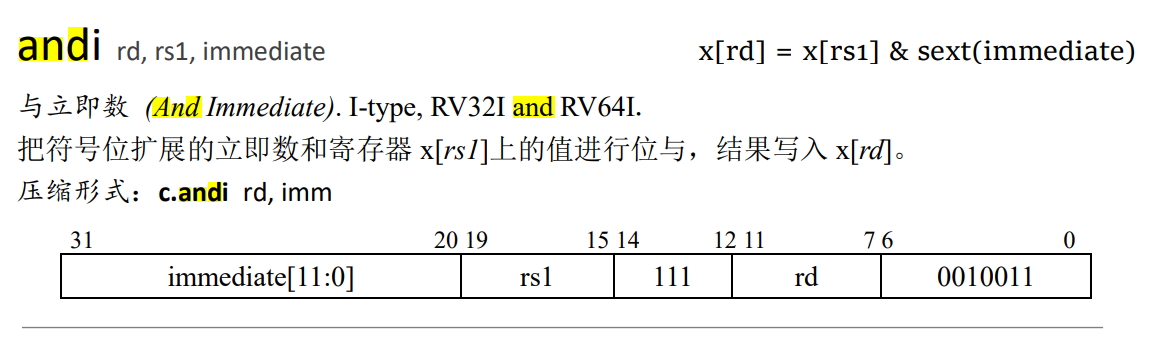
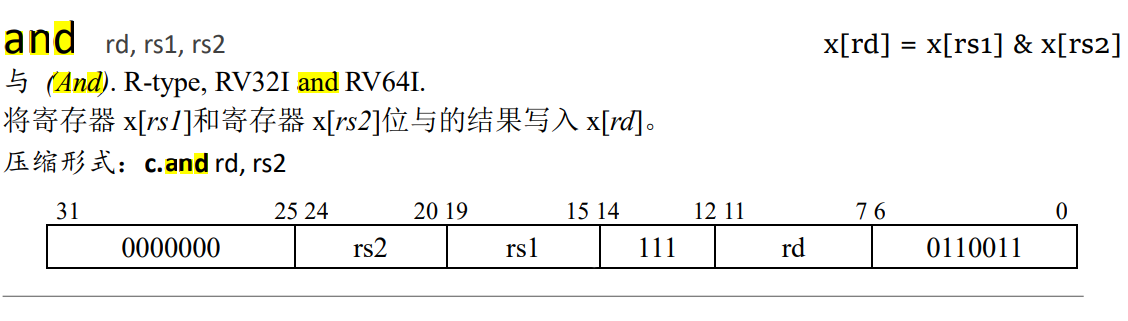


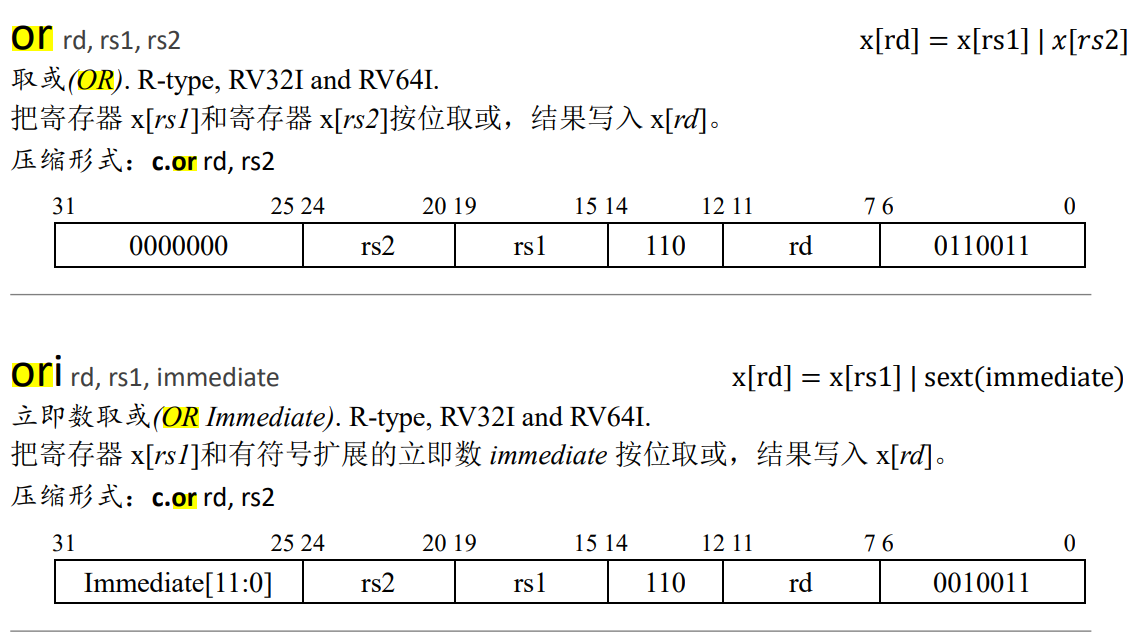


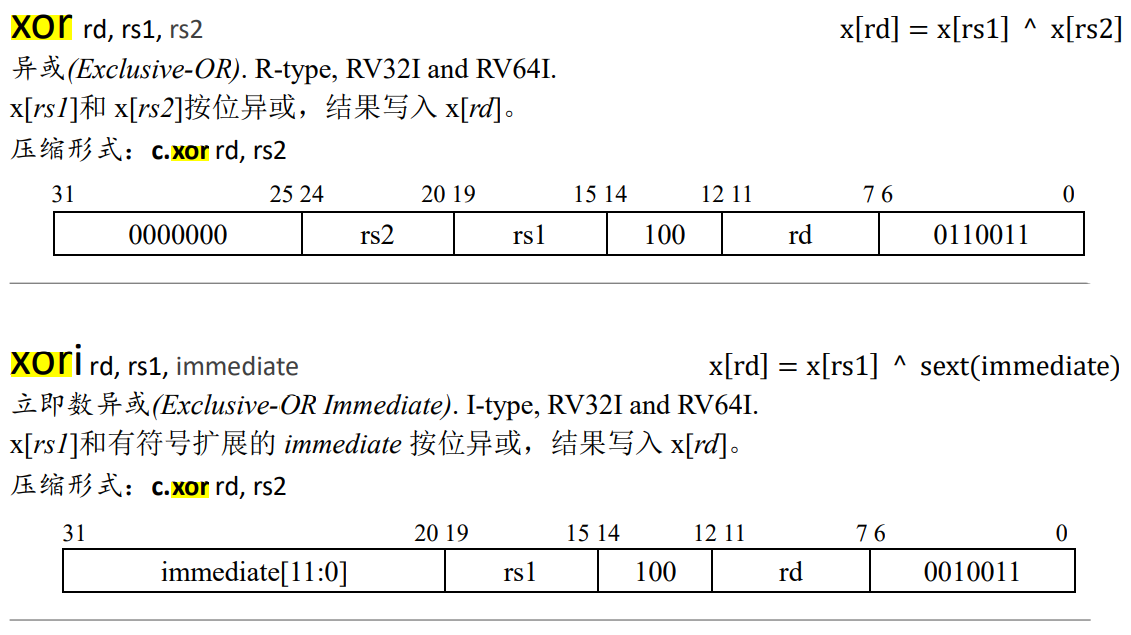


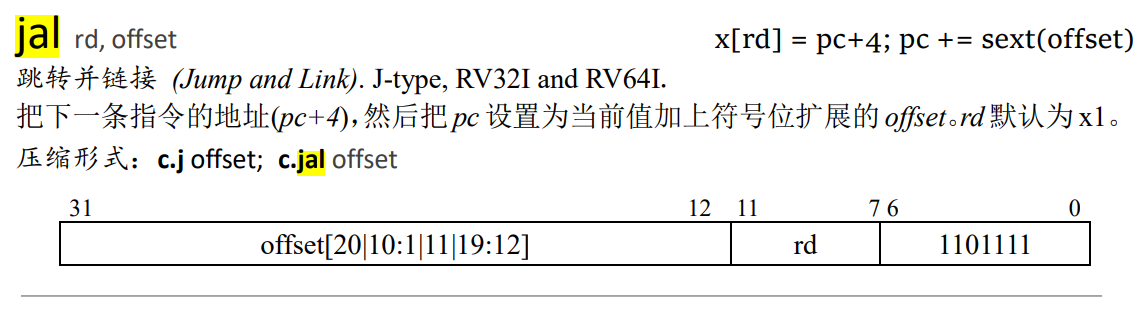


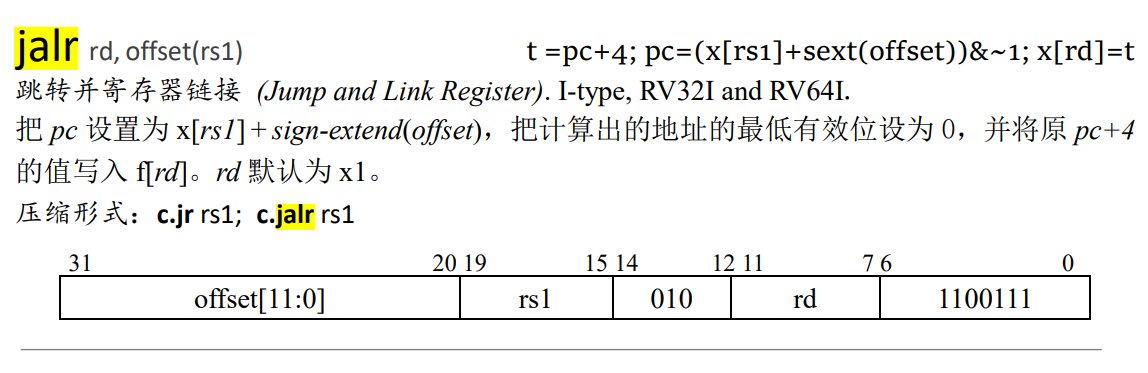


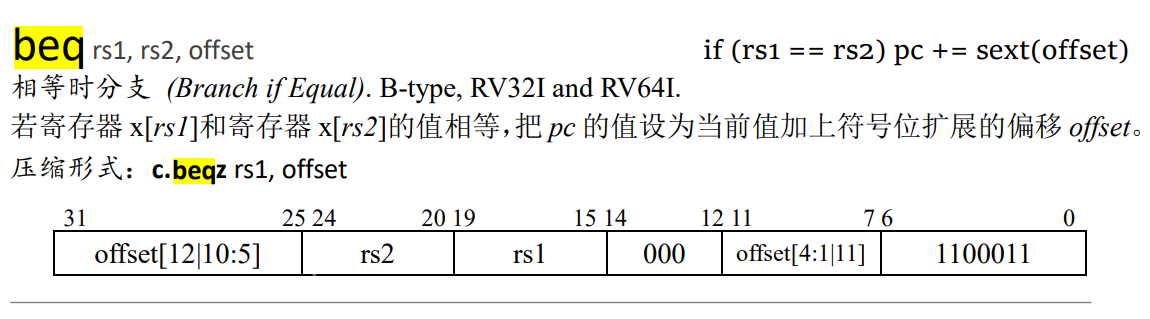


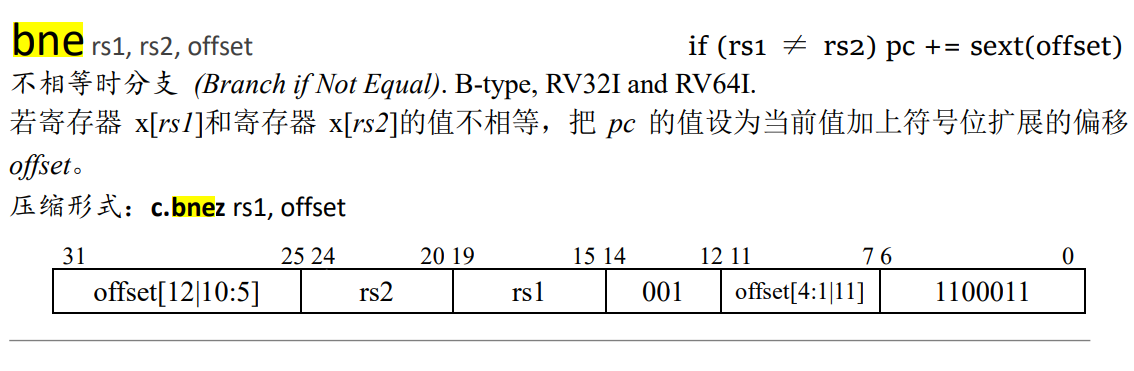


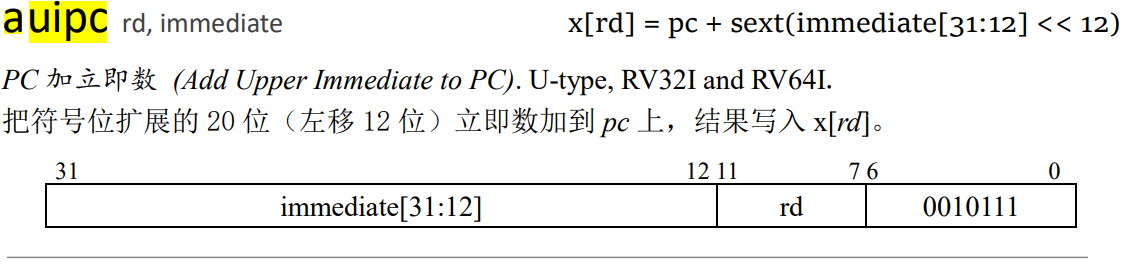












# 主要仪器设备

• HDL：Verilog、SystemVerilog

• IDE：Vivado

• 开发板：NEXYS A7

# 操作方法与实验步骤

## 操作方法

完成单周期CPU设计，并将其通过verilog电路实现，Run Simulation进行仿真并观察其波动情况。

## 实验步骤

4.2.1 数据通路设计

1. 根据lab10-1中给出的参考数据通路图完成设计

2. 根据提供的测试文件进行仿真测试

3. 提高提供的测试环境进行上板验证

4.2.2 控制模块设计

1. 完成控制单元模块设计

2. 结合数据通路，搭起单周期CPU

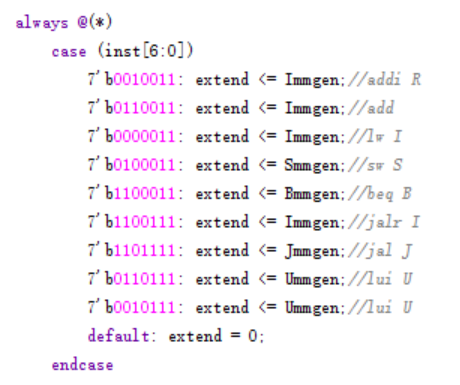
3. 使用lab10-1提供的仿真代码文件进行仿真测试

4. 使用lab10-1提供的测试环境进行上板验证

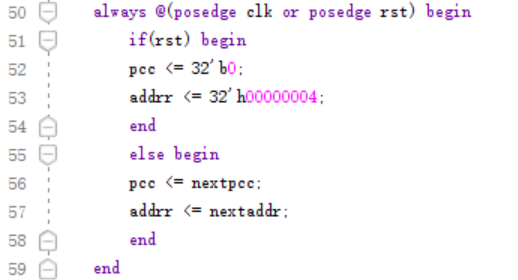
# 实验结果与分析

5.1 主要代码片段

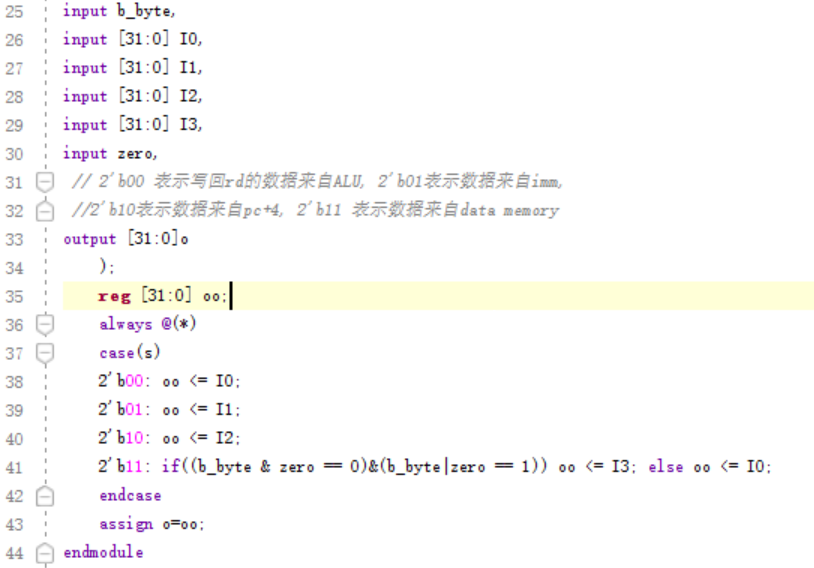
entend模块：根据指令解码，选择不同数字拓展类型。



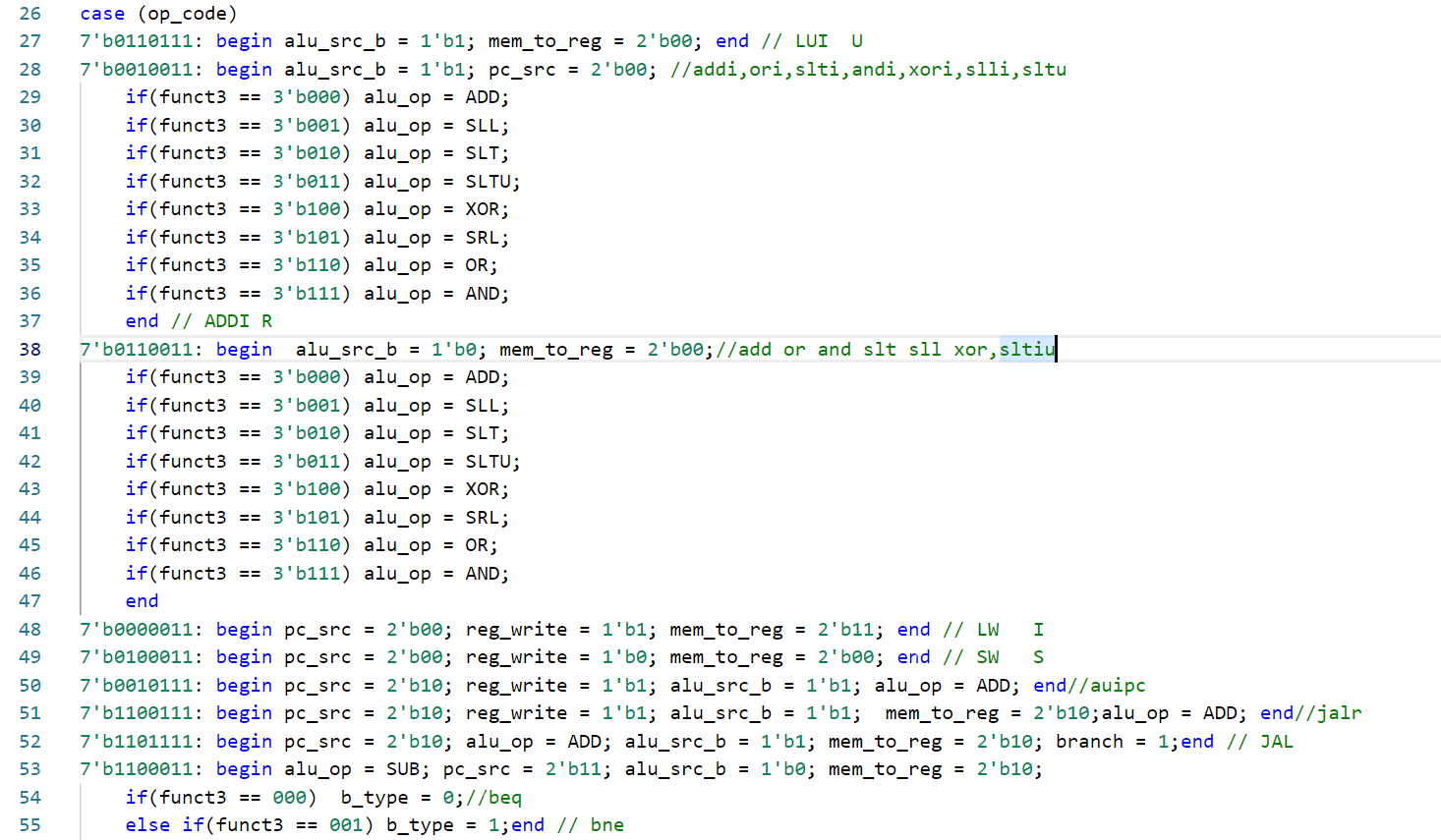
pc变化，每个时钟周期内，根据rst信号选择pc为pc+4或是初始地址。



多路选择器模块，根据Control模块传来的信号选择输出。

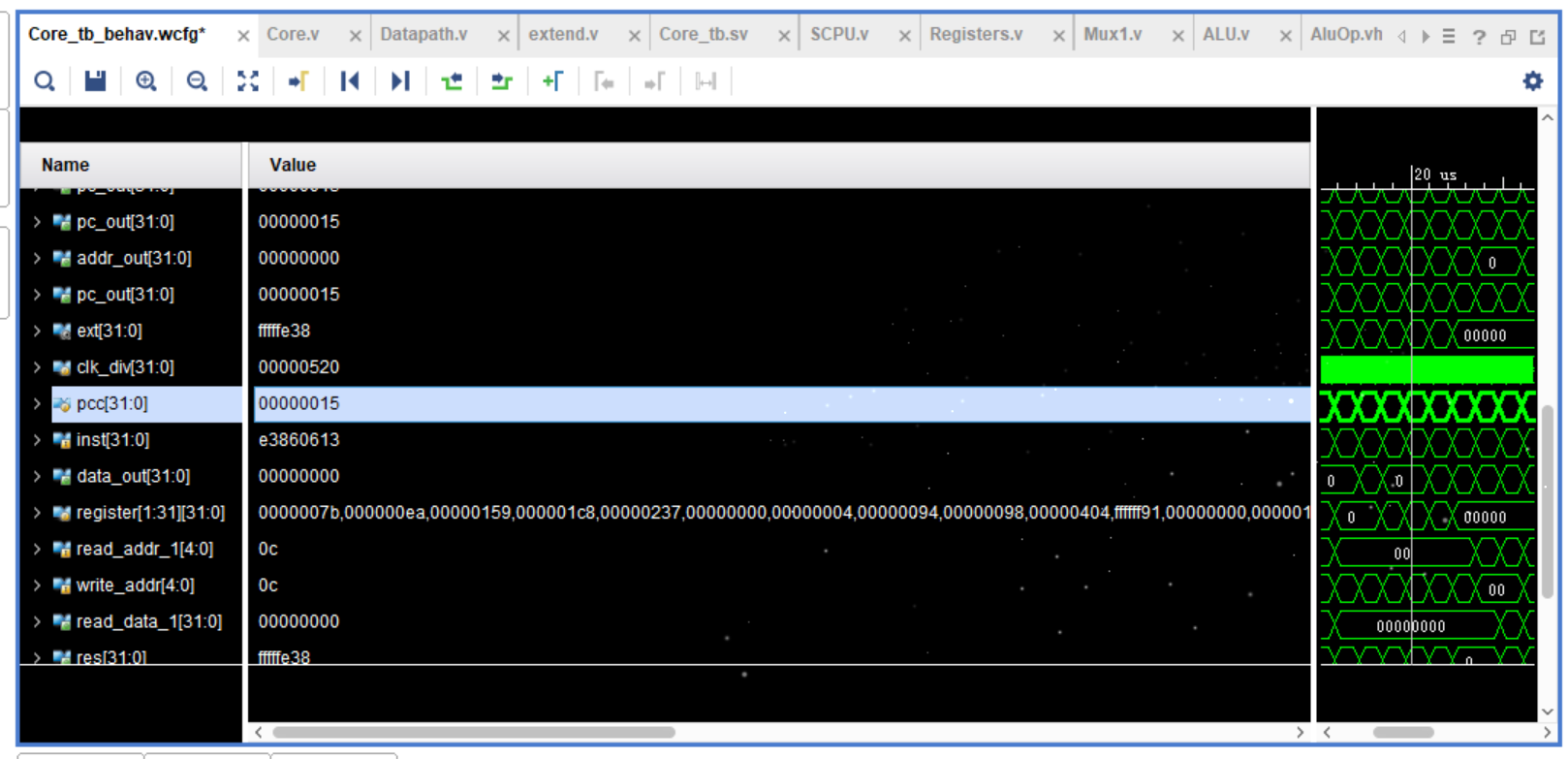


control模块，将指令解码，改变控制信号。



5.2 仿真模拟

首先rom连接lab\_10.coe, lw,sw,addi,bne指令成功后的波形变化。



自己编写的测试数据，验证bonus的指令。



上板验证，将pc当前指令地址，inst当前指令值， ld/st addr访问访存的地址， reg\_write\_data寄存器写回值在四个debug输出信号中输出。和汇编指令对应验证CPU正常运行。

