**深 圳 大 学 实 验 报 告**

**课 程 名 称： 计算机系统(3)**

**实验项目名称： 新增指令实验**

**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： 林浩晟 学号： 2022280310 班级： 01**

**实 验 时 间： 2024年12月7日**

**实验报告提交时间： 2024年12月9日**

**教务处制**

**一、 实验目标：**

了解RISC-V mini处理器架构，在其基础之上新增一个指令，完成设计并观察指令执⾏。

**二、实验内容**

1) 修改数据通路，新增指令comb rs1,rs2,rd采用R型指令格式，实现将rs1高16位和rs2低16位拼接成32位整数，并且保存到rd寄存器。

2) 在处理器上执行该指令，观察仿真波形，验证功能是否正确。

3）自行设计其他功能指令，并验证设计是否正确

**三、实验环境**

硬件：桌面PC

软件：Chisel开发环境

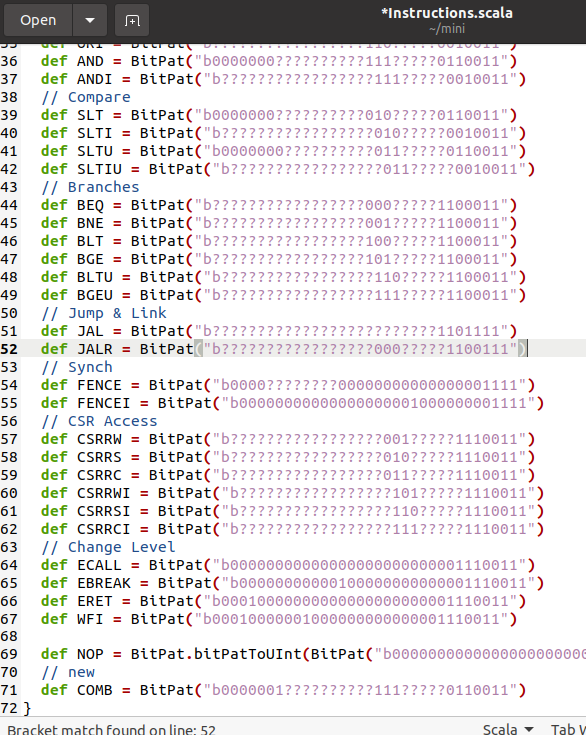
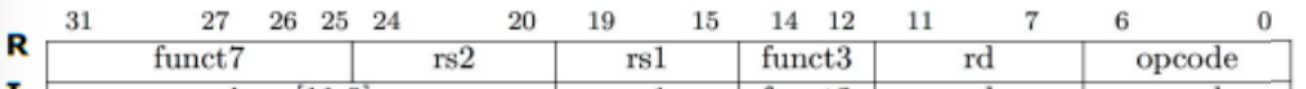
**四、****实验步骤及说明**

学习Chisel数据通路的Chisel描述，特别是指令译码部分和core核心代码。然后按照下面操作完成指令译码器的修改，以及数据通路的修改，

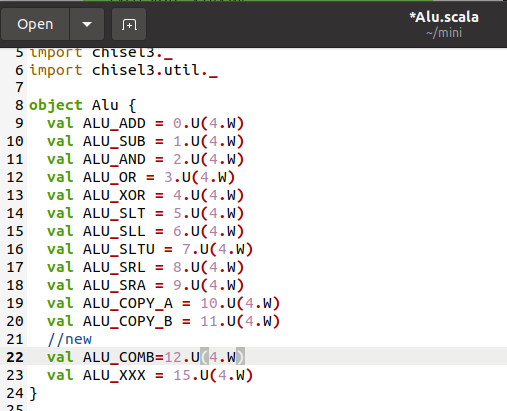
具体操作如下：按照参考文档完成comb指令的实现，自行设计新指令实现其功能并验证。

1. 在Instrutcions.scala文件中添加 comb 指令比特模式串；

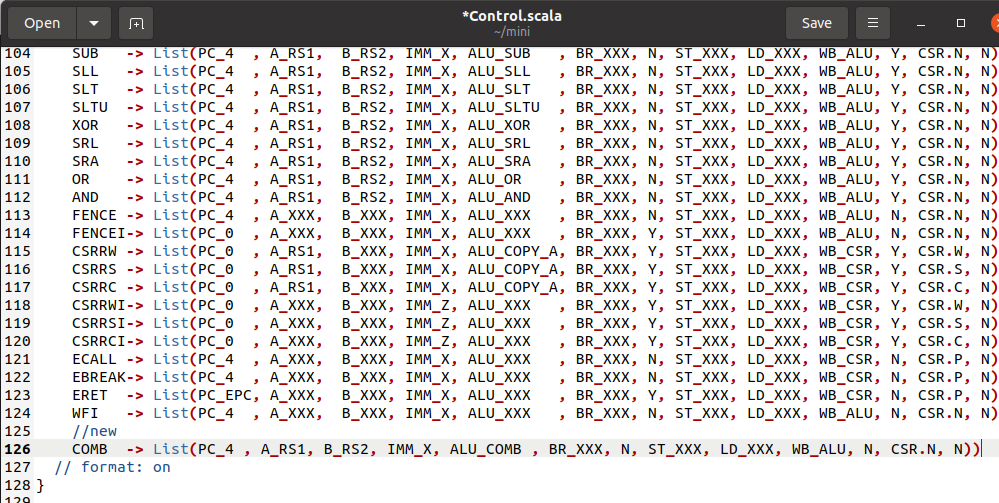
comb 为R型指令，riscv的R型指令格式如下：



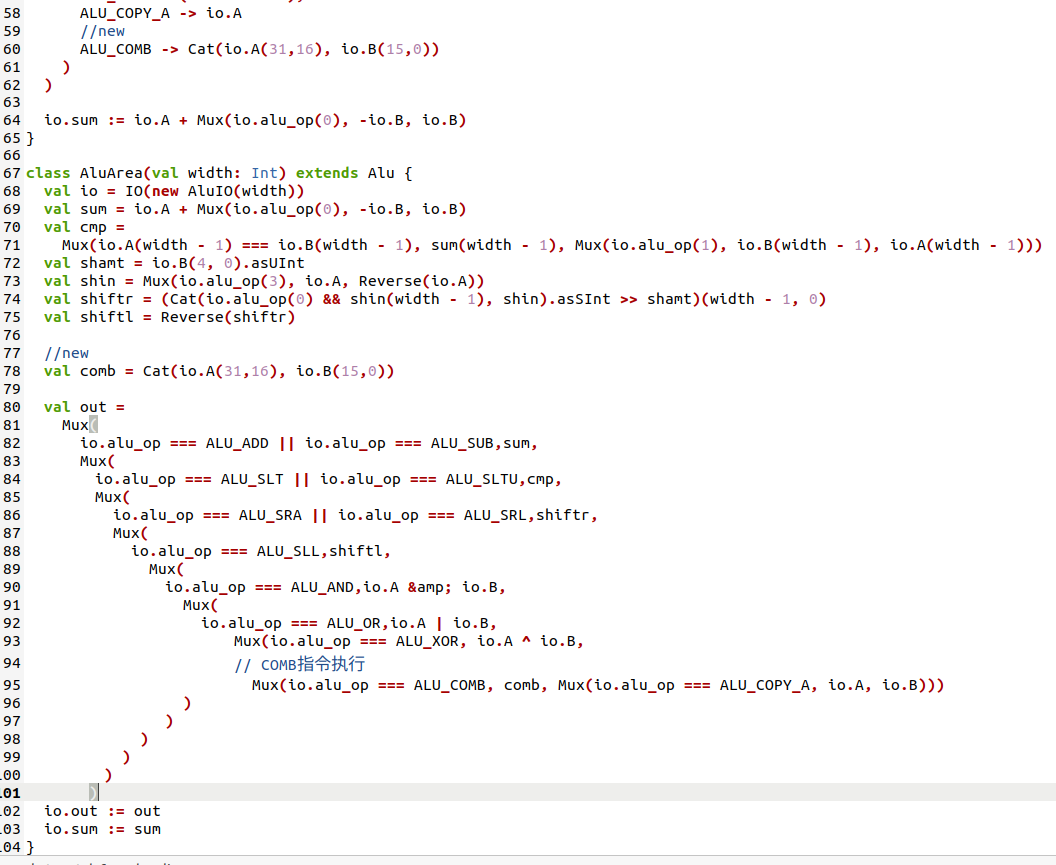
1. comb 指令需要在ALU中将rs1高16位和rs2低16位拼接成32位整数，因此需要在Alu.scala文件中添加常量 ALU\_COMB，让译码器可以译码出正确的信号，如下：



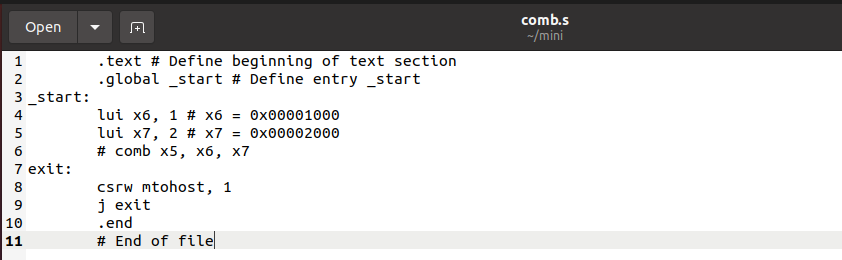
1. 接下来为 comb 指令添加对应的译码映射comb 指令执行后pc需要加4，并将从寄存器文件中读取的数据rs1和rs2进行拼接操作，然后 将ALU输出的拼接结果写回到寄存器文件中；



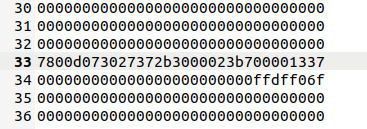
1. 在Alu.scala文件添加将rs1高16位和rs2低16位拼接成32位整数的操作；



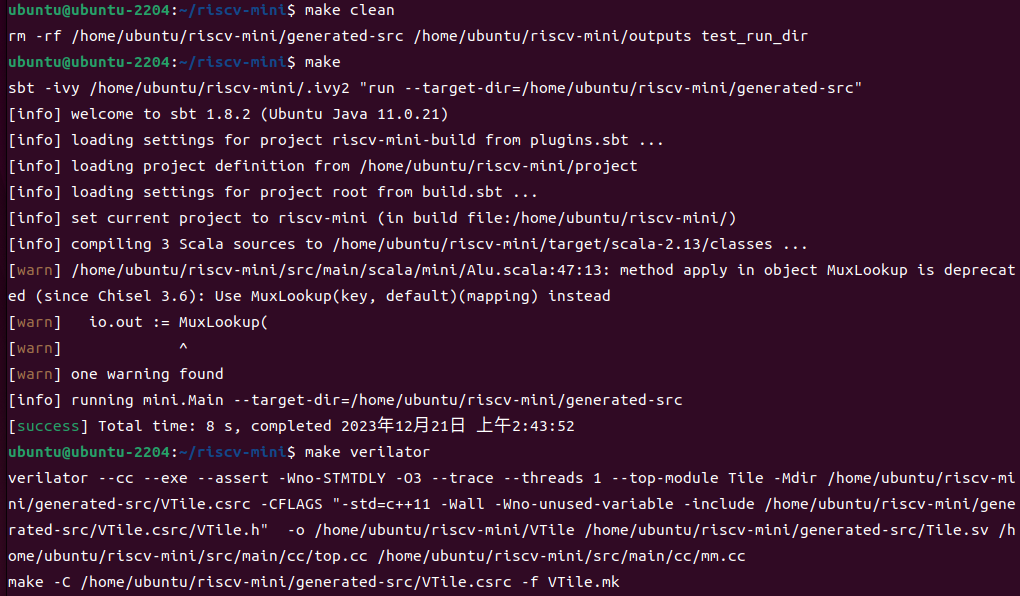
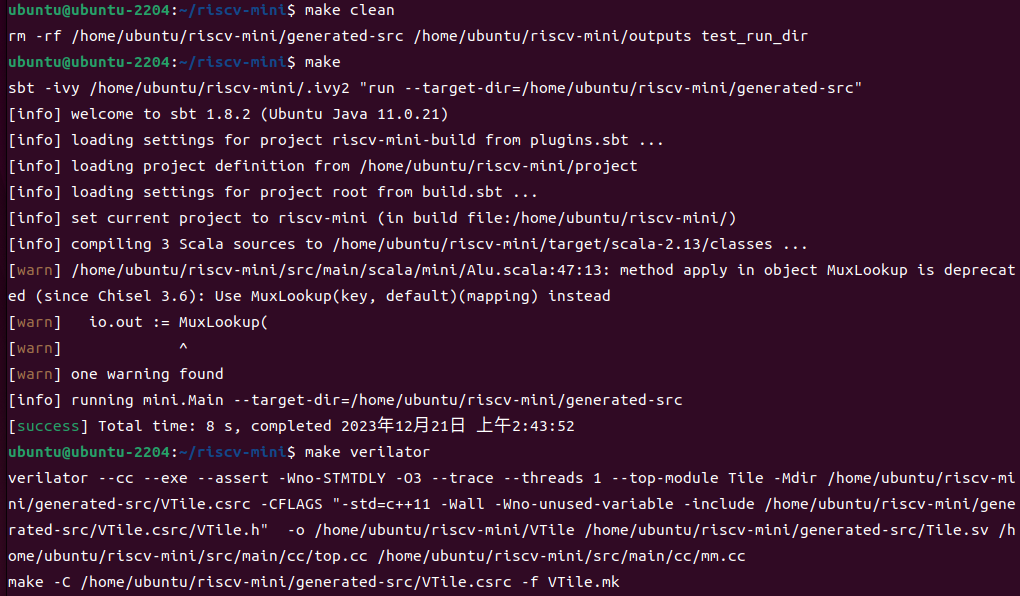
1. 创建comb.s文件，编写以下汇编语言；



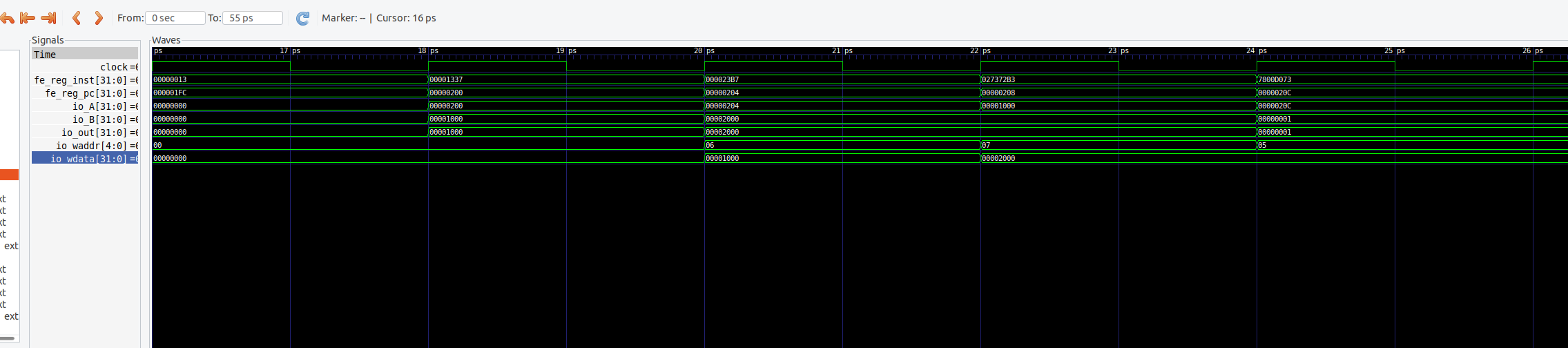
1. 将文件转换为十六进制文件，如下，指令 comb x5, x6, x7 对应的机器码的十六进制为 0x027372b3 ，因采用小端法存储，故需将其插入到上两条指令对应的机器码之前，如下图所示：



1. 分别执行make clean、make、make verilator指令，得到波形文件comb.vcd；



1. 打开comb.vcd，可以得到如下波形：



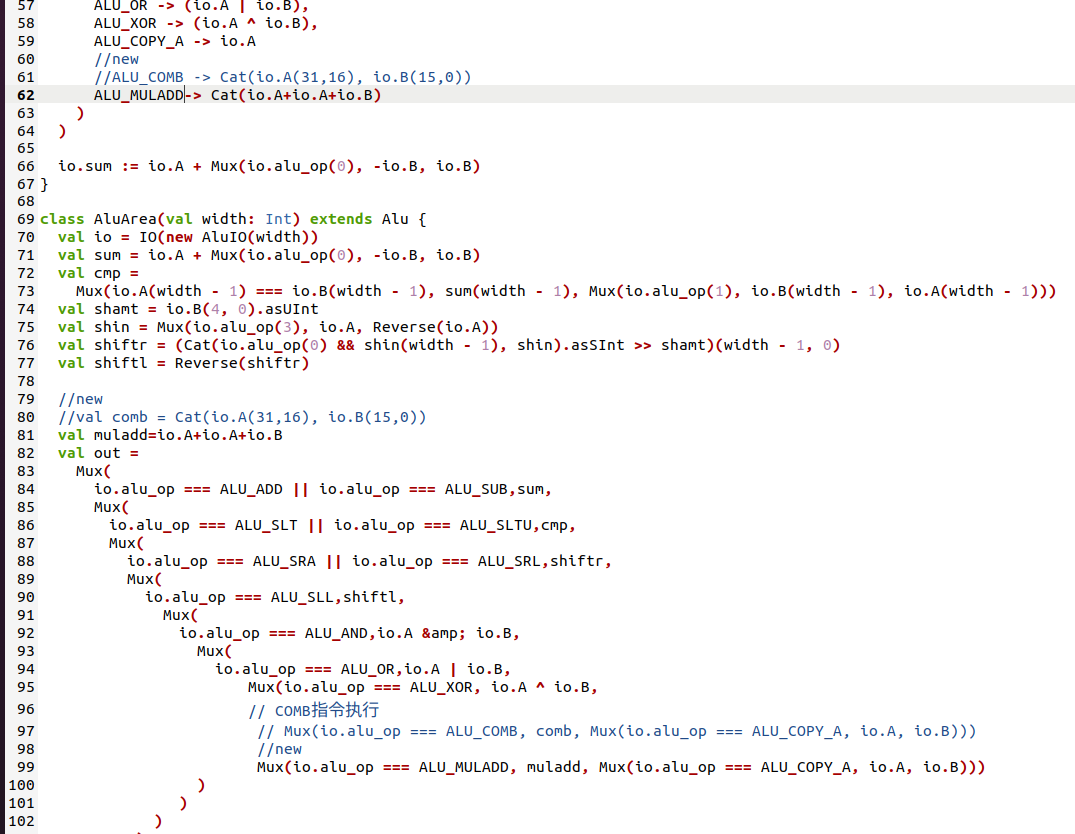
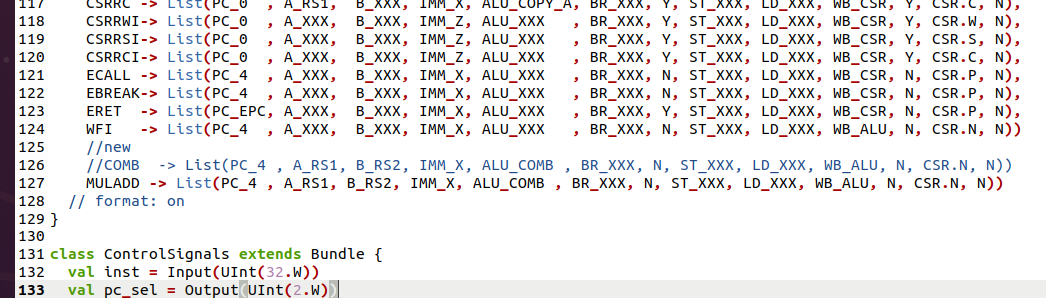
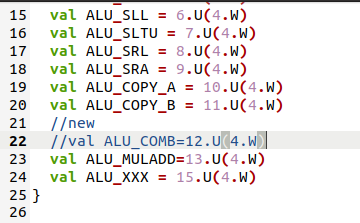
程序中 3 条主要的指令的十六进制如下。

|  |  |  |
| --- | --- | --- |
| 指令 | 十六进制机器码 | 说明 |
| lui x6, 1 | 0x00001337 | x6 = 0x00001000 |
| lui x7, 2 | 0x000023b7 | x7 = 0x00002000 |
| comb x5, x6, x7 | 0x027372b3 | x5 = cat(x6(31:16), x7(15:0)) = 0X00002000 |

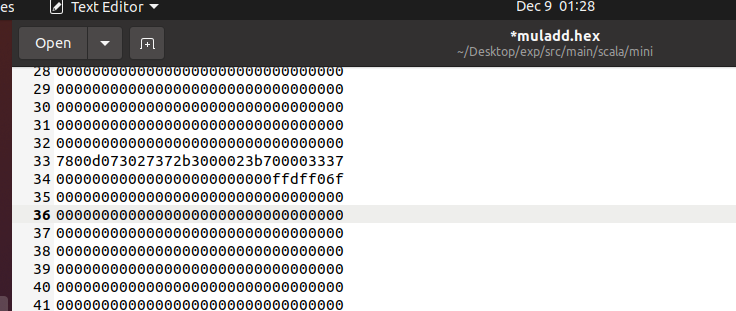
可以观察到程序读取x6和x7的值后进行拼接，结果 0x000020000写回x5 。

1. 这里自行设计了一个指令：newadd rd, rs, rt；功能为：R型指令，用以读取寄存器rs和 rt的值，计算2\*rs+rt，再将结果写回寄存器rd。

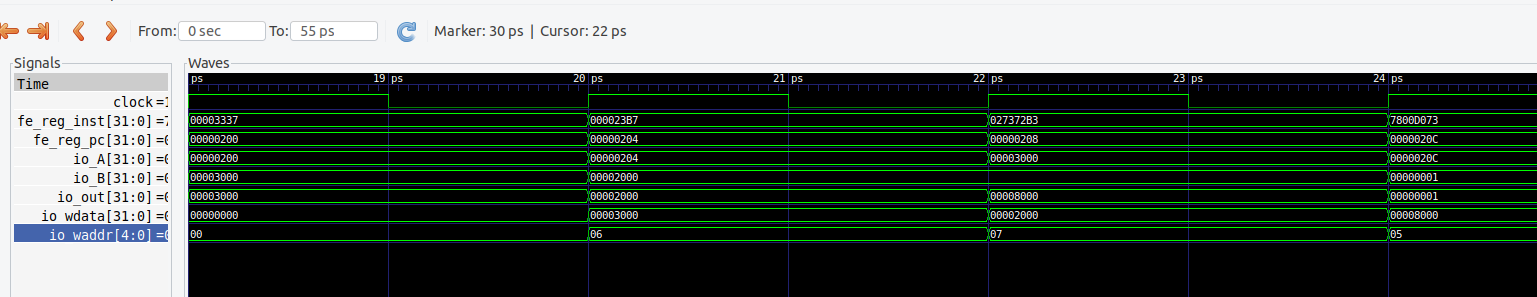
这里使用与comb相同的比特模式，随后按照comb相同的操作，分别调整译码常量、译码映射、指令执行等，如下图：



1. 在muladd.hex中找到指令luix6,3和指令luix7,2的机器码对应的十六进制，指令muladd x5,x6,x7的十六进制机器码为0x027372b3，由于采用小端法存储，将其插至上述两指令之前，如下图：



1. 同comb指令，执行make clean、make、make verilator指令后可以得到对应的vcd文件，打开后发现程序读取到x6和x7的值后，计算了2\*x6+x7的值，最后将结果写回x5。



其中程序中主要的指令的十六进制如下。

|  |  |  |
| --- | --- | --- |
| 指令 | 十六进制机器码 | 说明 |
| lui x6, 3 | 0x000033337 | x6 = 0x00003000 |
| lui x7, 2 | 0x000023b7 | x7 = 0x00002000 |
| newadd x5, x6, x7 | 0x027372b3 | x5 = 2 \* x6 + x7 = 0x00008000 |

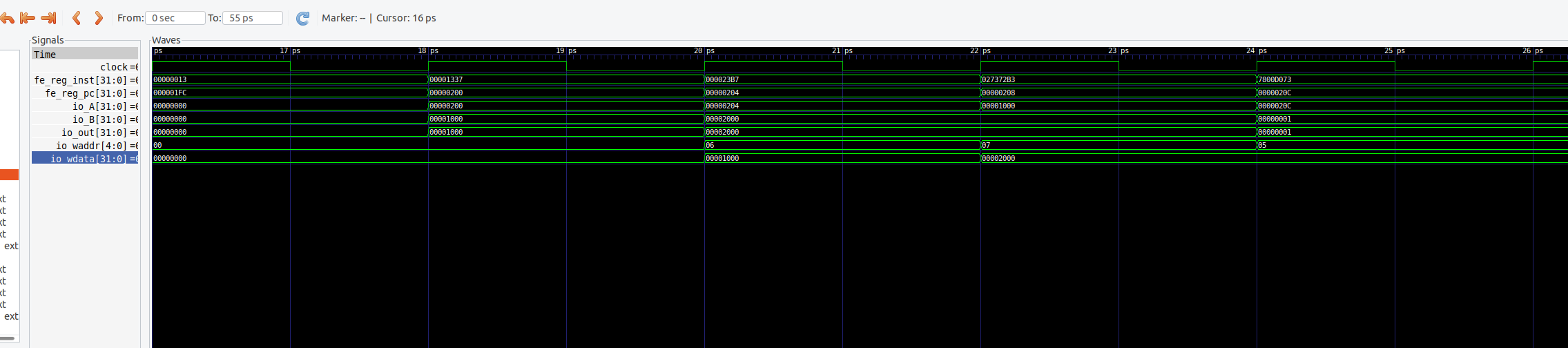
**五、实验结果、**

1．Comb指令

程序中3条主要的指令的十六进制如下图：

|  |  |  |
| --- | --- | --- |
| 指令 | 十六进制机器码 | 说明 |
| lui x6, 1 | 0x00001337 | x6 = 0x00001000 |
| lui x7, 2 | 0x000023b7 | x7 = 0x00002000 |
| comb x5, x6, x7 | 0x027372b3 | x5 = cat(x6(31:16), x7(15:0)) = 0X00002000 |

波形如下图所示：

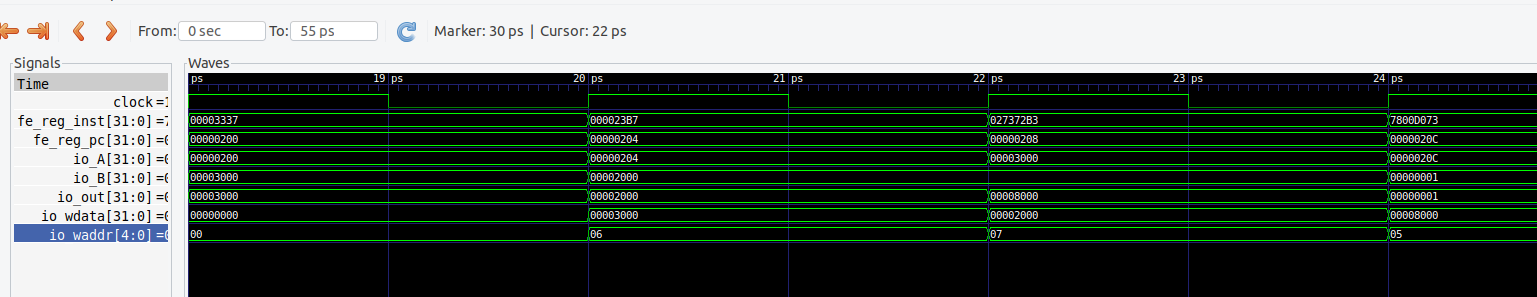


2. Muladd指令

程序中主要的指令的十六进制如下。

|  |  |  |
| --- | --- | --- |
| 指令 | 十六进制机器码 | 说明 |
| lui x6, 3 | 0x000033337 | x6 = 0x00003000 |
| lui x7, 2 | 0x000023b7 | x7 = 0x00002000 |
| newadd x5, x6, x7 | 0x027372b3 | x5 = 2 \* x6 + x7 = 0x00008000 |

波形如下图所示：



**六、实验总结与体会**

实验成功完成。

实验体会：

本次实验旨在通过修改RISC-V mini处理器架构以支持自定义指令，深化对RISC-V结构的理解，并验证对指令集架构的掌握。

在实验里面，给数据通路增加了comb指令，实现了rs1高16位与rs2低16位的拼接，涉及到ALU和寄存器文件的调整。

通过仿真工具观察新指令执行，加深了对数字逻辑和处理器内部结构的理解。

仿真波形图的分析锻炼了我的验证与调试技能，对处理器设计和软硬件协同开发至关重要。这次实验不仅让我学到了数字逻辑设计的实际应用，还提高了对计算机体系结构的理解，为未来深入研究计算机体系结构和处理器设计打下了坚实基础。

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