中国科学院大学计算机组成原理（研讨课）

实 验 报 告

学号：2021K8009925006 姓名：冯浩瀚 专业：计算机科学与技术

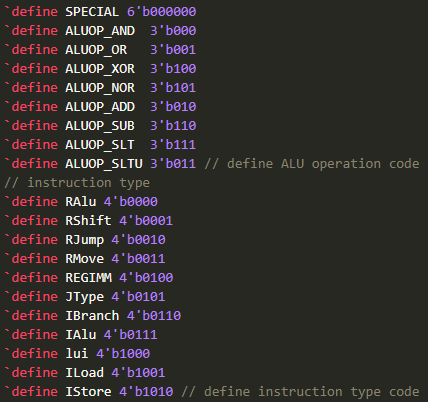
实验序号：02 实验名称：简单功能型处理器设计

注1：撰写此Word格式实验报告后以PDF格式保存SERVE CloudIDE的/home/serve-ide/cod-lab/reports目录下（注意：reports全部小写）。文件命名规则：prjN.pdf，其中“prj”和后缀名“pdf”为小写，“N”为1至4的阿拉伯数字。例如：prj1.pdf。PDF文件大小应控制在5MB以内。此外，实验项目5包含多个选做内容，每个选做实验应提交各自的实验报告文件，文件命名规则：prj5-projectname.pdf，其中“-”为英文标点符号的短横线。文件命名举例：prj5-dma.pdf。具体要求详见实验项目5讲义。

注2：使用git add及git commit命令将实验报告PDF文件添加到本地仓库master分支，并通过git push推送到SERVE GitLab远程仓库master分支（具体命令详见实验报告）。

注3：实验报告模板下列条目仅供参考，可包含但不限定如下内容。实验报告中无需重复描述讲义中的实验流程。

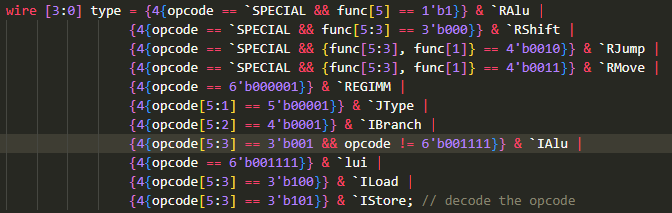
1. 逻辑电路结构与仿真波形的截图及说明（比如关键RTL代码段{包含注释}及其对应的逻辑电路结构图{自行画图，推荐用PPT画逻辑结构框图，复制到word中}、相应信号的仿真波形和信号变化的说明等）
2. 宏定义

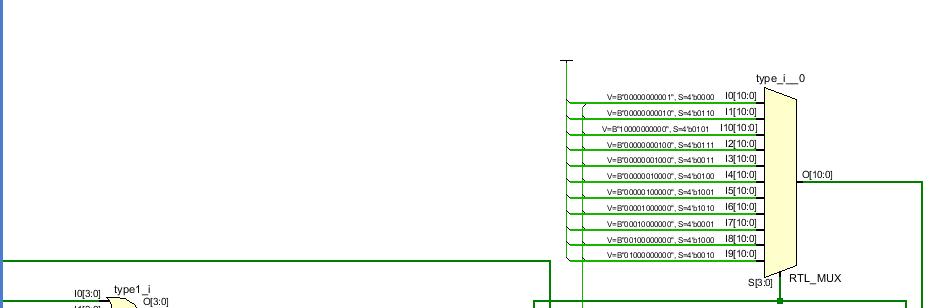


目的是提高代码的可读性

1. type分类

对指令进行分类，方便后续电路的逻辑控制

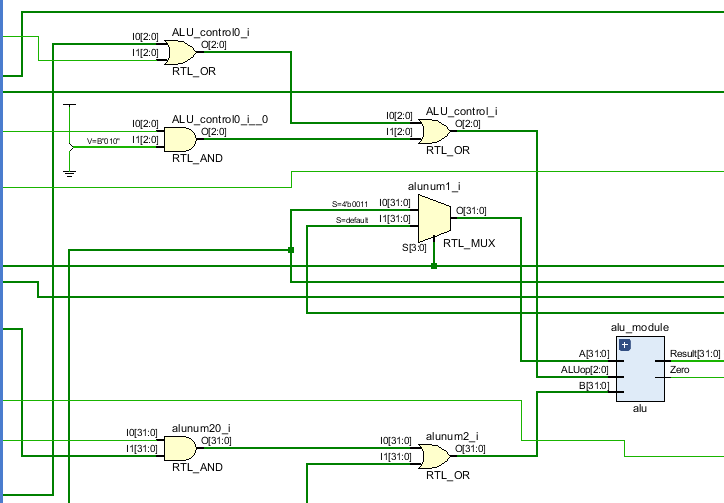




1. Alu Control

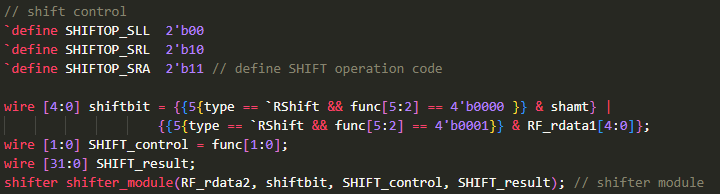
根据不同的指令所需的运算类型以及数据来源，使用选择器对alu模块的输入进行选择，输出运算结果和Zero指标

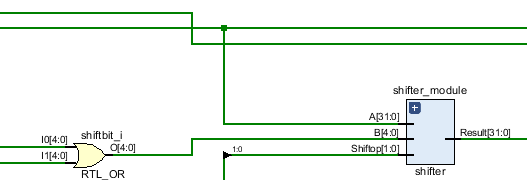




1. Shifter control

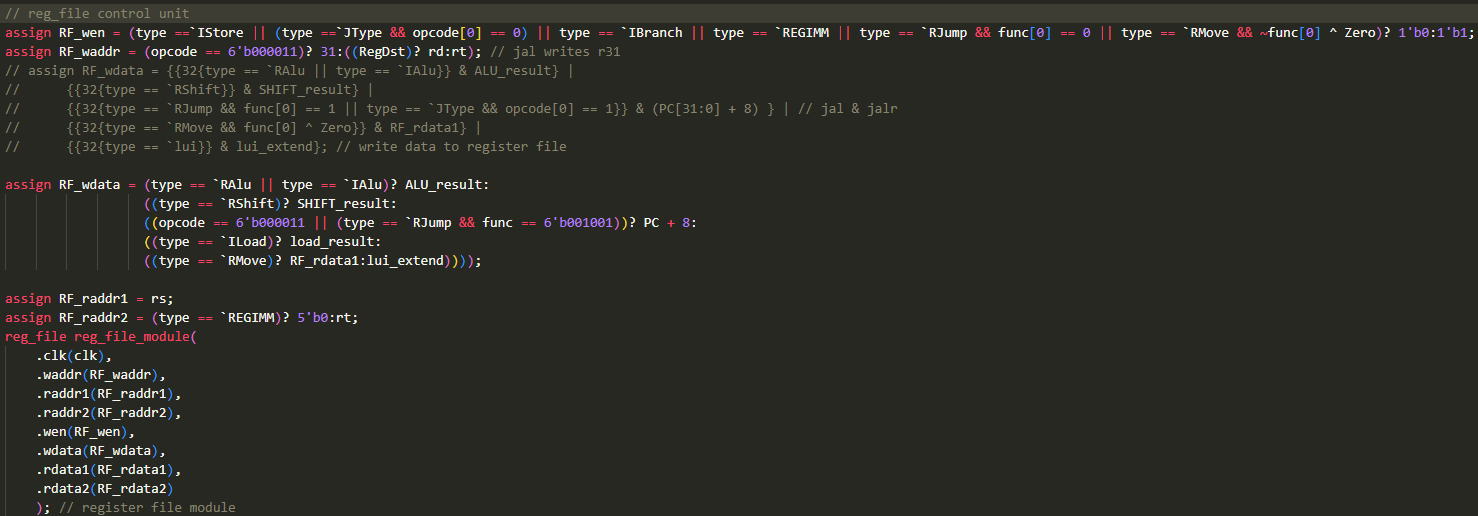
根据不同的指令所需的移位类型以及数据来源，使用选择器对shifter模块的输入进行选择，输出移位结果。

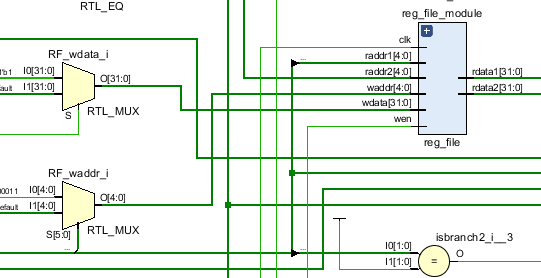




1. Reg\_File Control

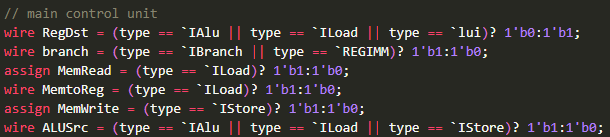
根据不同的指令所需读出或写入的寄存器位置，对reg\_file模块进行例化，输出读出数据或进行写入操作。





1. Main Control

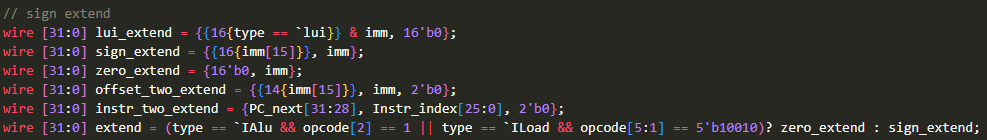
一些关键的控制信号，控制alu, PC, Reg\_File等模块



这些信号在逻辑电路结构图中分布较散，不一一截图

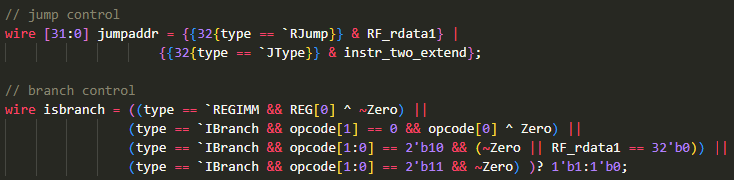
1. sign extend

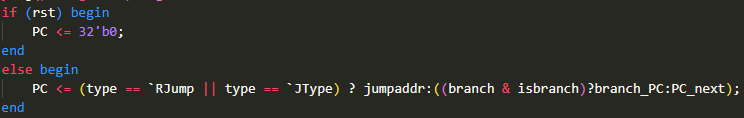
处理MIPS指令中存在一些对立即数的位拓展操作

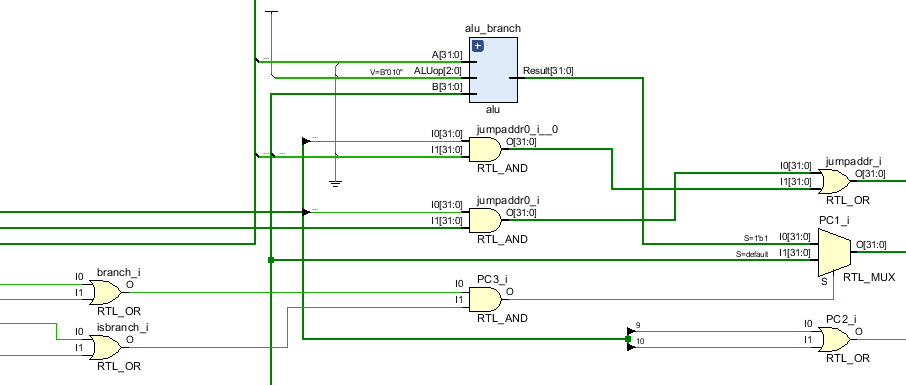


1. jump & branch

这两类指令的共同点在于需要更改PC的值

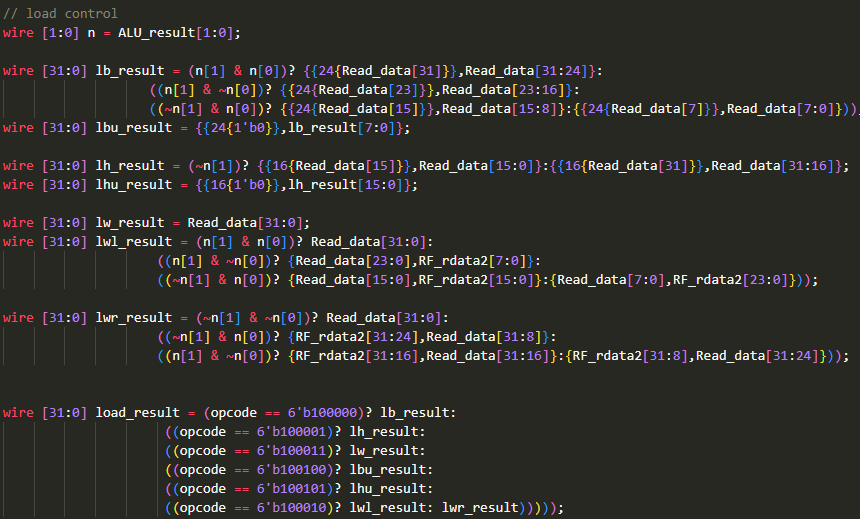


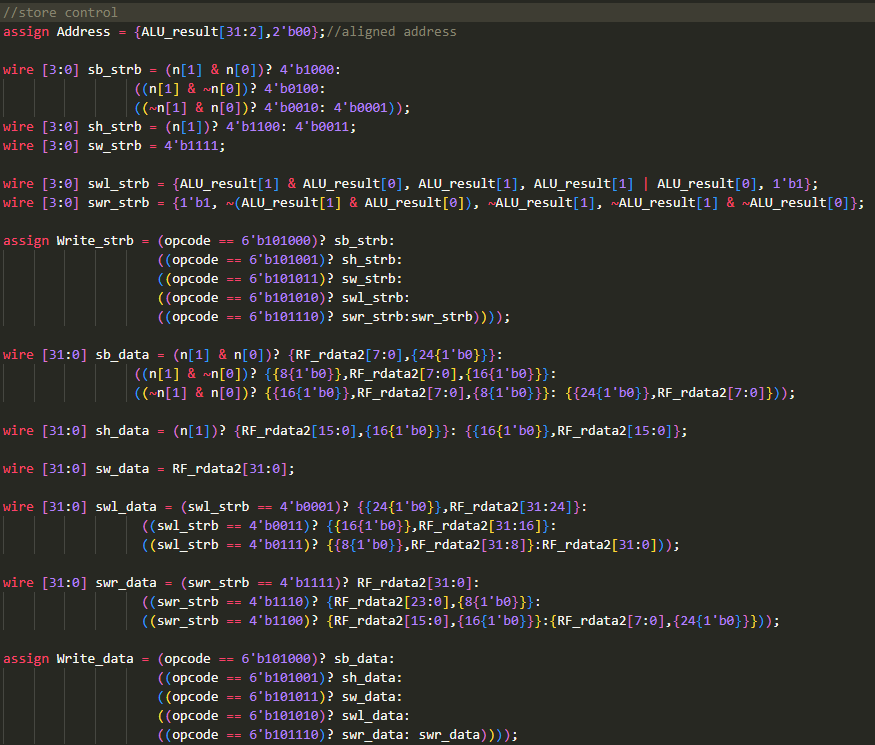


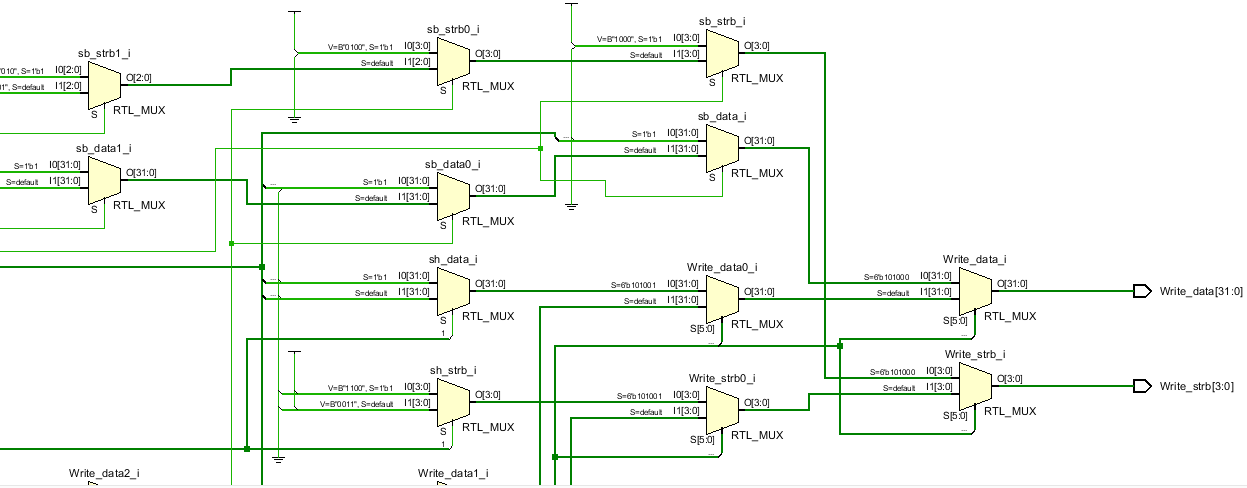


1. load & store

这两类指令的代码难点在于对数据灵活地进行分割、连接等操作







1. 实验过程中遇到的问题、对问题的思考过程及解决方法（比如RTL代码中出现的逻辑bug，逻辑仿真和FPGA调试过程中的难点等）
2. 控制信号的选择器的条件不完善，漏掉了某些指令的情况。根据发生错误时的指令类型以及其他信号的波形推断出错的位置。这类错误发生的最多，因为一次性将45条指令的操作类型、操作条件、有关寄存器的位置等代码编写完整很困难，只能一步一步修缮完整，所以这类bug是本次实验的主旋律。
3. 编写shifter模块时算术右移不能使用“>>>”，应该采用一种将原操作数进行逻辑右移之后再与11…100…0（1的个数与移位位数相同）按位或的方法，手动移位。
4. 在课后，你花费了大约 20 小时完成此次实验
5. 对于此次实验的心得、感受和建议（比如实验是否过于简单或复杂，是否缺少了某些你认为重要的信息或参考资料，对实验项目的建议，对提供帮助的同学的感谢，以及其他想与任课老师交流的内容等）
6. 本次实验较困难，首先需要对单周期处理器的原理、流程图完全理解，之后对代码框架的搭建也考验了我的大局观。我认为这个项目应该从整体入手，将alu，reg\_file，shifter，PC等模块写出来，然后从细节入手，参考指令集处理这些模块的输入与cpu的输入之间的关联。
7. 本次实验的参考资料较为齐全，但是在一些细微之处存在一些误导性的话语，不过因此导致的程序bug很好找出。例如，“P03-简单功能型处理器设计-v6.pdf”中的第25页指出，“所有R-Type指令都会产生寄存器写（wen=1）”。然而笔者发现bhv\_sim中要求jr和movz与movn的wen并不总是1。
8. 总之，感谢老师们的辛勤付出。