**计算机体系结构实验课程第 四 次实验报告**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 实验名称 | 对alu实验进行修改 | | | 班级 | 张金老师 |
| 学生姓名 | 袁田 | 学号 | 2314022 | 指导老师 | 董前琨 |
| 实验地点 | 实验楼A306 | | 实验时间 | 2025.4.10 | |

1. **实验目的**

(1)熟悉MIPS指令集中的运算指令，学会对这些指令进行归纳分类。

(2)了解MIPS指令结构。

(3)熟悉并掌握ALU的原理,功能和设计。

(4)进一步加强运用verilog语言进行电路设计的能力。

(5)为后续设计cpu的实验打下基础。

1. **实验内容说明**

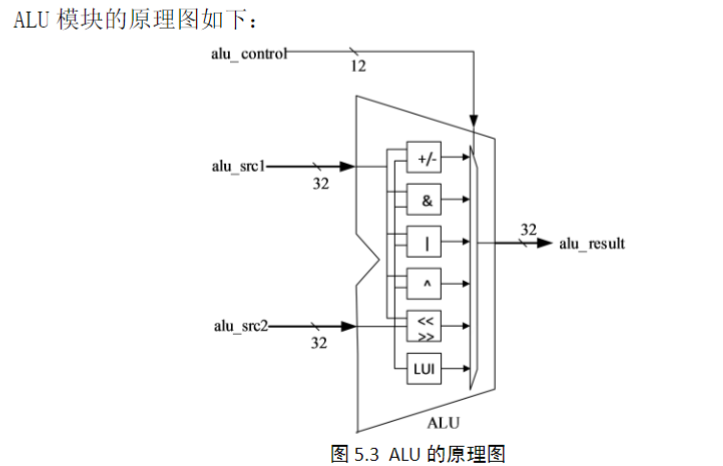
(1)对原始实验进行复现，在实验箱上面进行验证，确定实验准确性

(2)将原有的操作码进行位压缩，调整操作码控制信号位宽为4位。

(3)操作码调整成4位之后，在原有11种运算的基础之上，补充3种不同类型的运算（要求一种大于置位比较，一种位运算，一种自选），需要上实验箱或仿真验证计算结果。

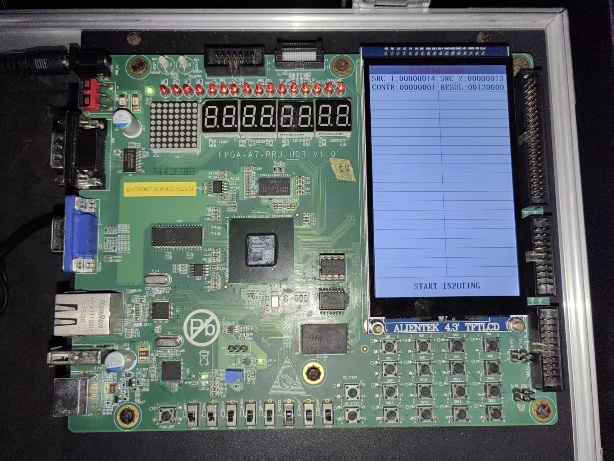
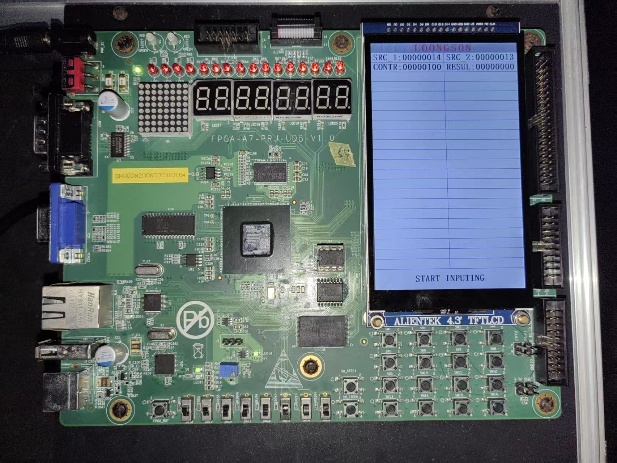
(4)注意改进实验上实验箱验证时，操作码应该已经压缩到4位位宽。

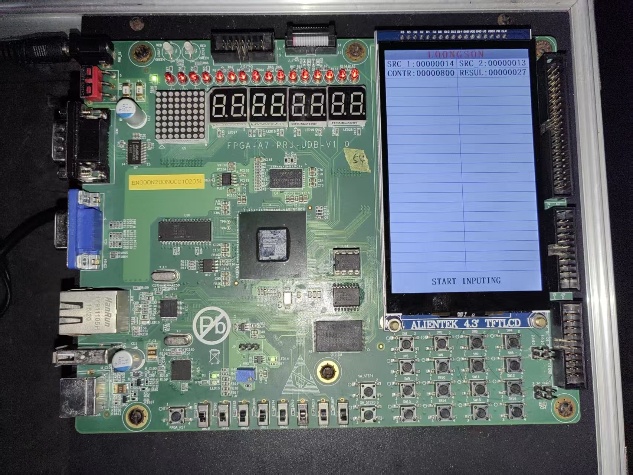
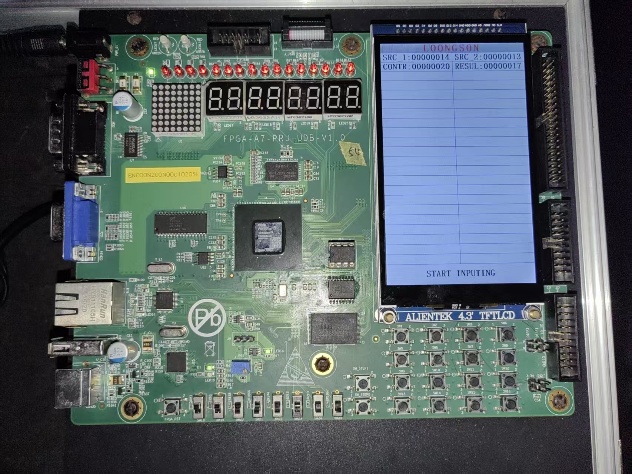
1. **实验原理图**

****

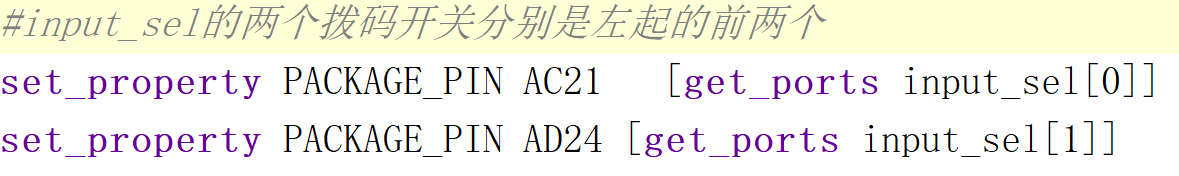
1. **实验步骤**

(1)对原实验进行复现

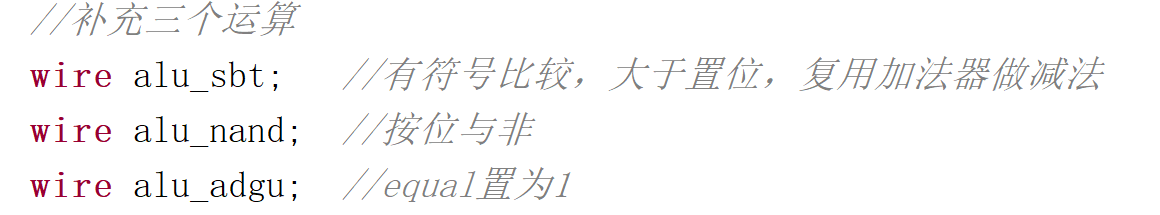


这里实现了：无符号比较(小于置位)，由于操作数一为0x14，操作数二为0x13，由于操作数一大于操作数二，则比较结果为大于，此时结果数为0，符合预期；将操作数二的低十六位移到高十六位，实验结果同样符合预期，结果数为0x130000

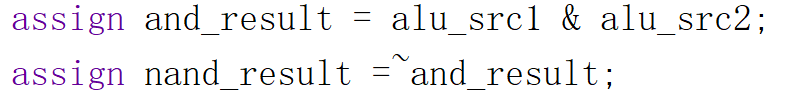
此处实现了按位或功能：操作数一为0x14(即00010100)，操作数二为0x13(即00010011)，则结果数为0x17(即00010111)，符合实验预期；加法操作：结果数为操作数一二相加，符合实验要求情况。

另外，为了便于调整input\_sel，将控制该数的拨码开关改为前两个开关

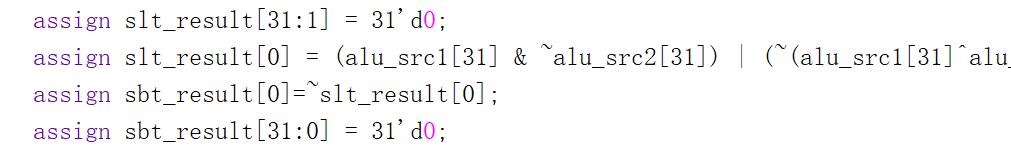
(2)增加功能共有三个：有符号比较的大于置位；按位与非；判断两操作数是否相等



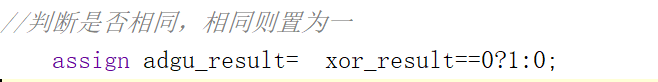
(3)对这三个功能的实现：



按位与非的结果为按位与操作结果数取反获得；



有符号的大于置位，很自然可以看出，相对于有符号的小于置位，大于置位的结果情况恰好相反，即大于置位结果为小于置位结果取反；

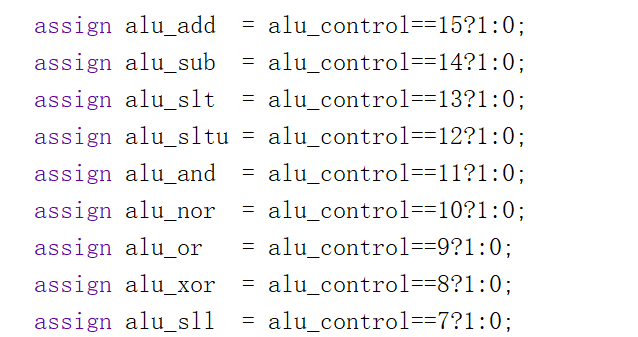


由于按位异或的结果只在两操作数相同时为0，其他情况下均不为0，则只需要判断按位异或的结果是否为零，则可判断两操作数是否相等

(4)由于操作码调整成4位后的修改部分

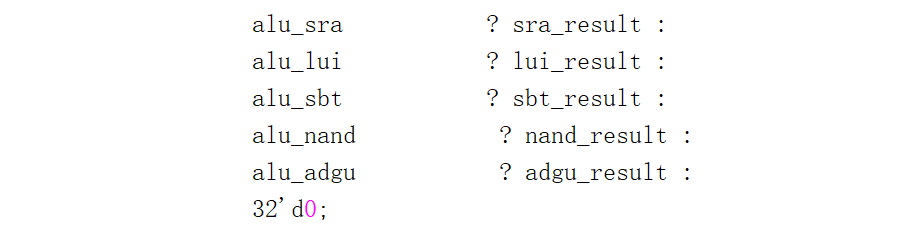
首先，控制信号位宽需要改变：需要将input [11:0] alu\_control 改为 input [3:0] alu\_control

其次，需要修改ALU控制信号，独热码的赋值情况

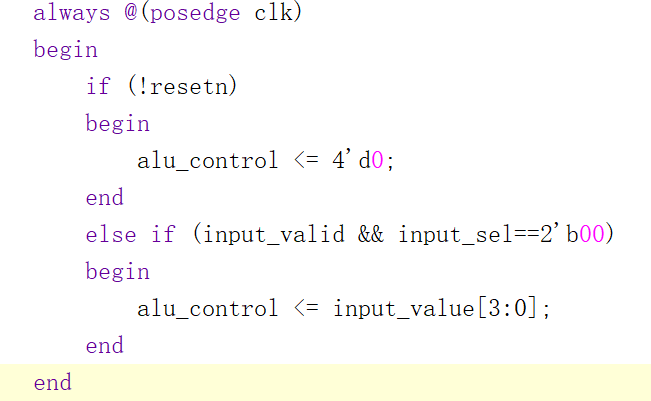


由于alu\_control为四位位宽的输入，取值为0-15共十六种情况，除去并未进行设计的为零的情况外，每一种赋值都代表一种运算，因此上述语句可进行判定

然后，需要修改最后的判定语句，确保最终结果alu\_result的值与选择情况对应(添加三个新加的功能)



最后，调整上层模块，操作码应该已经压缩到4位位宽

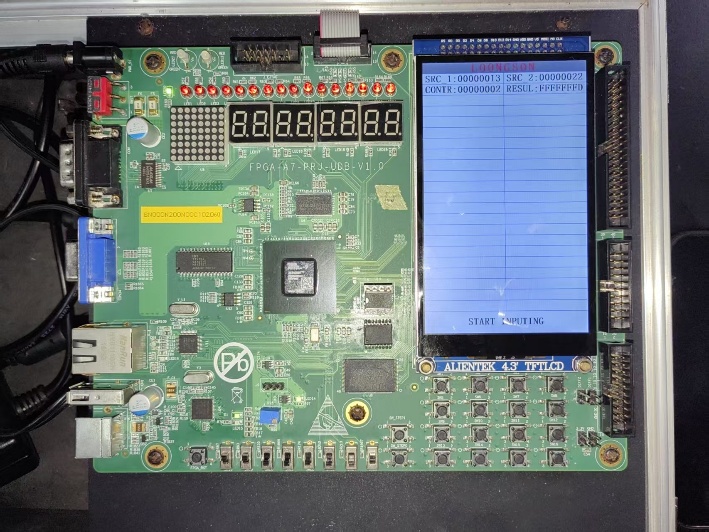


1. **实验结果分析**

(1)alu\_control=3，此时执行的运算为比较中的大于置位，此时操作数一为0x13，操作数二为0x25，由于操作数一小于操作数二，则结果为零

****

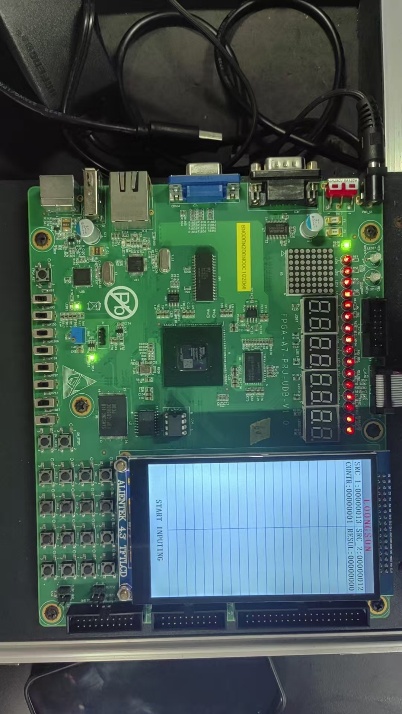
(2) alu\_control=2，此时执行的运算为与非操作，此时操作数一为0x13(即00010011)，操作数二为0x22(即00100010)，由于操作数一按位与上操作数二结果为00000010，进行取反后则结果为FD，由于高位均为零取反后则为F，符合实验设计



(3) alu\_control=1，此时执行的运算为判断两个操作数是否相等，当相等时结果为1，此时操作数一为0x13，操作数二为0x13，由于操作数相等，则结果为1

****

而当操作数不相等时，结果为零

****

1. **总结感想**

通过本次实验，对于计算机中的运算是如何进行的，并且理解了在汇编代码中某些指令在计算机内部的运行过程，并知道了MIPS指令集中的运算主要是分为算术运算，逻辑运算(比如按位运算，左右移位)，存储和读取等运算