组成原理实验课程第_4_次实验报告

实验名称	ALU 模块实现			班级	李涛
学生姓名	胡博浩	学号	2212998	指导老师	董前琨
实验地点	津南实验楼 A308		实验时间	2024.5.16	

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。

2、 实验内容说明

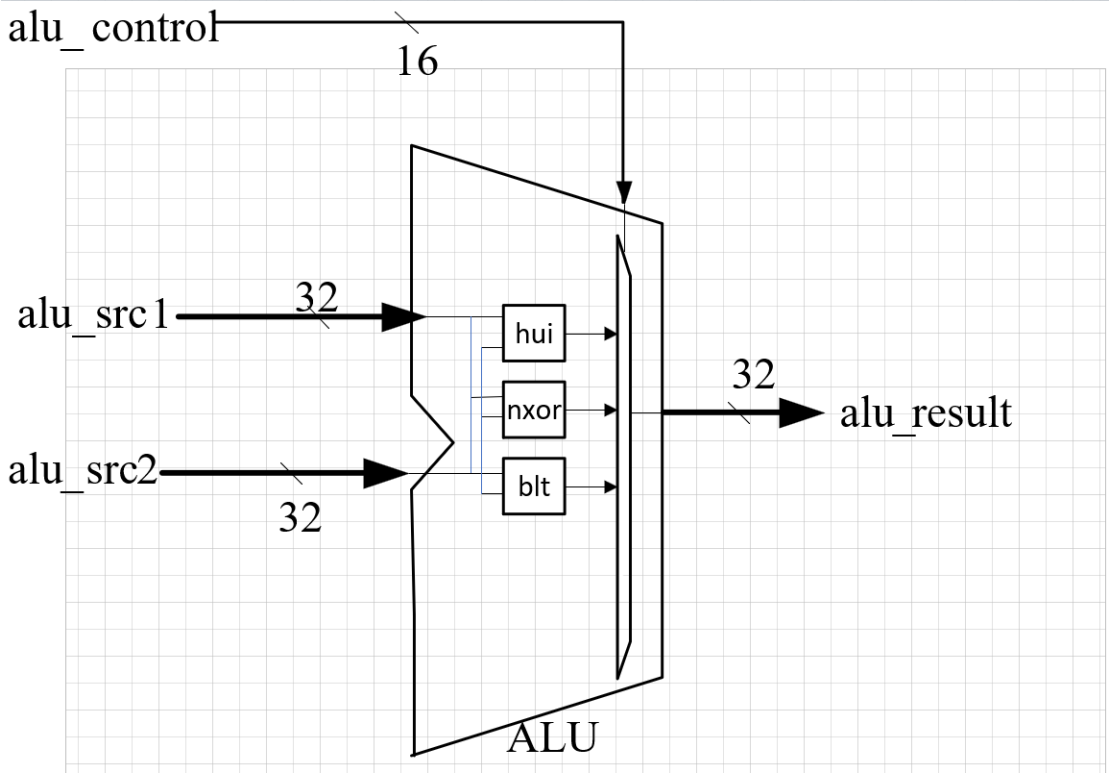
请结合实验指导手册中的实验四(ALU 模块实现实验)完成功能改进,实现一个能够完成更多运算的 ALU,注意以下几点:

1、原始代码中只有表 5.1 中的 11 种运算,请另外补充至少三种不同类型运算(比较运算、位运算、数据加载运算等)。

2、实验报告中原理图参考 5.3,只画出补充的运算部分即可,报告中展示的结果应位补充运算的计算结果。

3、本次实验报告不需要有仿真结果,但需要有实验箱上箱验证的照片,同样,针对照片中的数据需要解释说明。

实验原理图



将控制信号缩减为 4 位、以表示 15 种运算；添加三个新运算：hui---低位加载、nxor---按位同或、blt---有符号大于则置位。

3、 实验步骤

本次实验只需要修改两个文件，alu.v 和 alu_display.v 文件。

（1）alu.v

a）修改 alu_control 的位宽，改为 4 位位宽

```
input  [3:0] alu_control,  // ALU控制信号                    将原先12位的控制信号压缩成了4位
```

b）修改独热码

首先添加三个控制信号

```
wire alu_hui; //低位加载
wire alu_nxor;//按位同或
wire alu_blt;//有符号大于则置位
```

然后减少位宽，改为 4 位二进制编码。

```
assign alu_add =(alu_control==4'b0001 ? 1 : 0);//1 为加法
assign alu_sub =(alu_control==4'b0010 ? 1 : 0); //2 为减法
assign alu_slt =(alu_control==4'b0011 ? 1 : 0); //3 为有符号比较
assign alu_sltu =(alu_control==4'b0100 ? 1 : 0); //4 为无符号比较
assign alu_and =(alu_control==4'b0101 ? 1 : 0); //5 为按位与
assign alu_nor =(alu_control==4'b0110 ? 1 : 0); //6 为按位或非
assign alu_or =(alu_control==4'b0111 ? 1 : 0); //7 为按位或
assign alu_xor =(alu_control==4'b1000 ? 1 : 0); //8 为按位异或
assign alu_sll =(alu_control==4'b1001 ? 1 : 0); //9 为逻辑左移
assign alu_srl =(alu_control==4'b1010 ? 1 : 0); //A 为逻辑右移
assign alu_sra =(alu_control==4'b1011 ? 1 : 0); //B 为算术右移
assign alu_lui =(alu_control==4'b1100 ? 1 : 0); //C 为高位加载
assign alu_hui =(alu_control==4'b1101 ? 1 : 0);//D 为低位加载
assign alu_nxor =(alu_control==4'b1110 ? 1 : 0);//E 为按位同或
assign alu_blt =(alu_control==4'b1111 ? 1 : 0);//F 为有符号大于则置位
```

最后添加三个变量，用于存储三个运算的结果。

```
wire [31:0] hui_result;
wire [31:0] nxor_result;
wire [31:0] blt_result;
```

c）编写三个新运算的代码

对于按位同或，直接把异或结果取反就行

```
assign nxor_result=~xor_result; //同或结果为异或取反
```

对于低位加载，和高位加载相反

```
assign hui_result = { 16'd0, alu_src2[15:0]};  //低位加载，与高位加载相反
```

对于有符号比较，大于则置位操作，利用真值表、并仿照小于置位操作编写

```
//blt 结果,有符号大于置位比较操作
assign blt_result[31:1]=31'd0;
assign blt_result[0]=(~alu_src1[31] & alu_src2[31])|(~(alu_src1[31]^alu_src2[31]) &~adder_result[31]&(adder_result!=32'b0));
```

d）把结果添加到最后的结果集中

```
assign alu_result = (alu_add|alu_sub) ? add_sub_result[31:0] :
                    alu_slt           ? slt_result :
                    alu_sltu          ? sltu_result :
                    alu_and           ? and_result :
                    alu_nor           ? nor_result :
                    alu_or            ? or_result  :
                    alu_xor           ? xor_result :
                    alu_sll           ? sll_result :
                    alu_srl           ? srl_result :
                    alu_sra           ? sra_result :
                    alu_lui           ? lui_result :
                    alu_hui           ? hui_result ://低位加载
                    alu_nxor          ? nxor_result ://按位同或
                    alu_blt           ? blt_result ://有符号数比较，大于置位
                    32'd0;
```

（2）alu_display.v

a）修改 alu_control 的位宽

```
reg   [3:0] alu_control;  // ALU控制信号
```

b）修改 alucontrol 的写入，对齐 4 位位宽

```
//当input_sel为00时，表示输入数控制信号，即alu_control
always @(posedge clk)
begin
    if (!resetn)
    begin
        alu_control <= 12'd0;
    end
    else if (input_valid && input_sel==2'b00)
    begin
        alu_control <= input_value[3:0];
    end
end
```

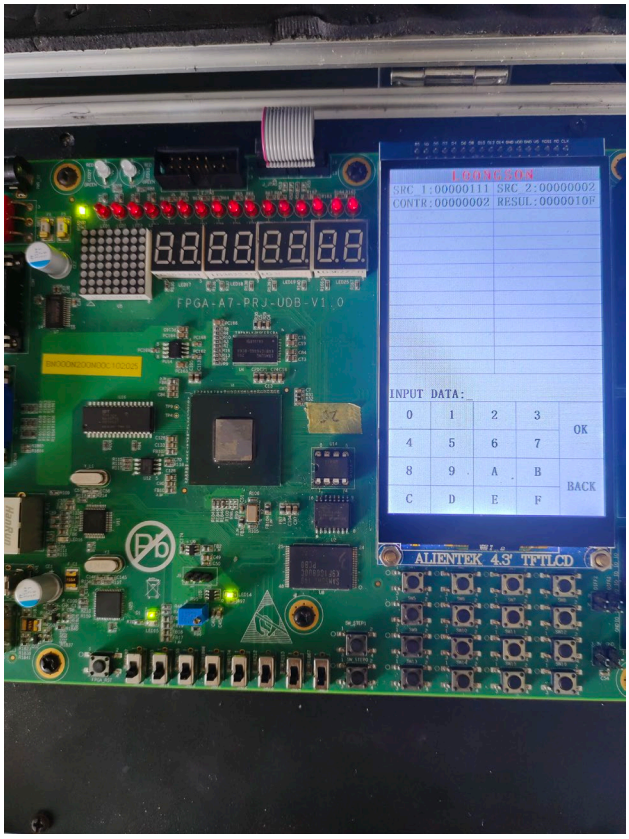
4、 实验结果分析

改进后支持的运算及其控制信号如下所示：

控制信号 CONTR	ALU 操作
0	无
1	加法
2	减法
3	有符号比较，小于置位
4	无符号比较，小于置位
5	按位与
6	按位或非
7	按位或
8	按位异或
9	逻辑左移
A	逻辑右移
B	算数右移
C	高位加载
D	低位加载
E	按位同或
F	有符号数比较，大于置位

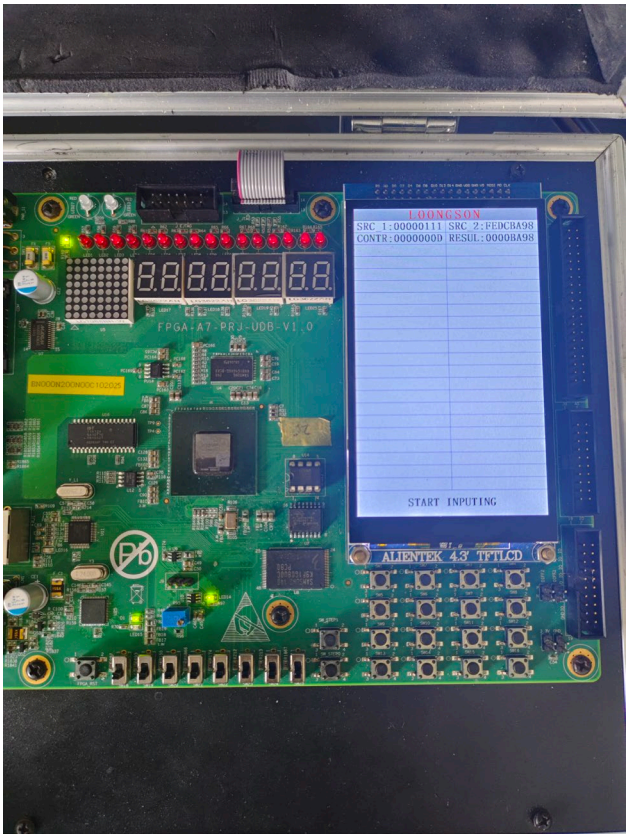
上箱实验验证

a）原始的减法运算



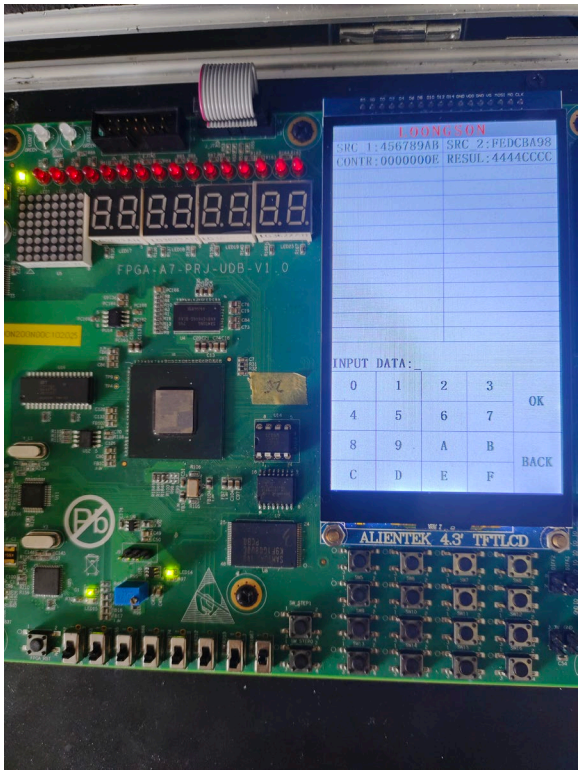
RESULT=0x111-0x2=0x10F，结果正确

b）新加的低位加载运算：



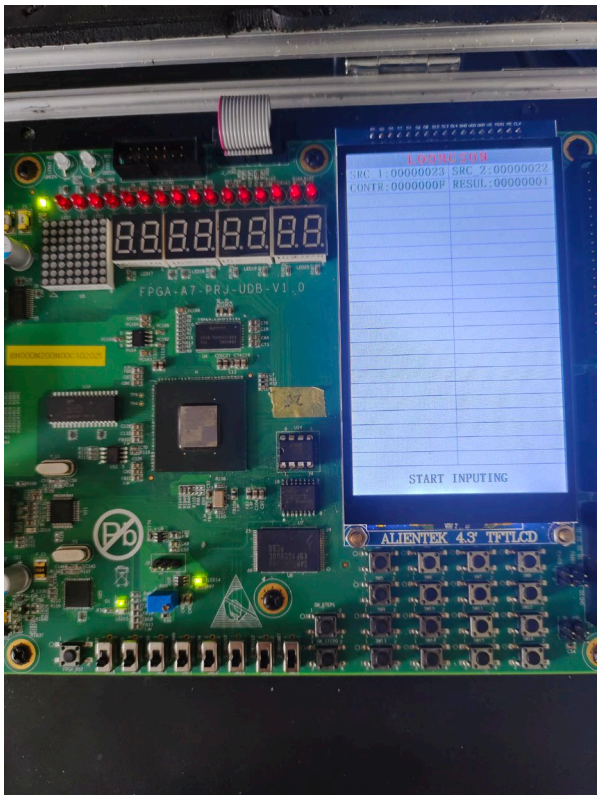
输入 SRC_2=FEDCBA98，RESULT=0000BA98，结果正确

c）新加的按位同或运算：

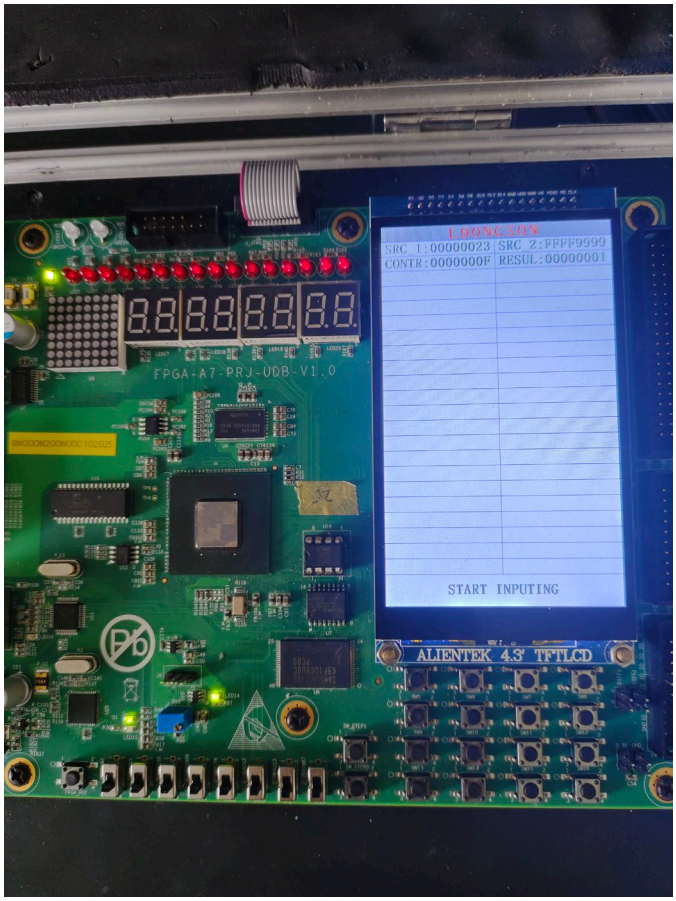


输入 456789AB、FEDCBA98，输出 4444CCCC，结果正确

d）新加的有符号大于则置位操作：



输入 23、22，输出 1，结果正确



输入 23、FFFF9999，输出 1，结果正确

综上所述，ALU 模块的实现和改进正常，实验成功！

5、 总结感想

完成这次 ALU 实验的改进让我受益匪浅。首先，通过对操作码进行位压缩和扩展运算种类的操作，我学会了如何在有限的位宽内灵活调整运算功能，这对我在数字电路设计中更好地利用有限的资源提高硬件性能具有极大的帮助。这种灵活性不仅增强了我的技术能力，也培养了我解决问题的能力，为我今后的工作和学习打下了坚实的基础。

其次，通过直接在实验箱上验证实验结果，我将理论知识与实际操作相结合，加深了对原理图和 Verilog 代码的理解。这种实践能力的提升不仅增强了我的实际操作能力，还培养了我分析和解决问题的能力。在未来的学习和工作中，我会继续保持对理论和实践的结合，不断提升自己的技术水平和解决问题的能力。这次实验的经历让我更加自信地迎接未来的挑战。