

计算机组成原理-实验三-多功能ALU设计实验

原创

ywm_up 于 2018-06-04 21:29:50 发布 31498 收藏 173

分类专栏: [计组](#) 文章标签: [计算机组成原理](#) [ALU运算器](#) [多功能ALU设计实验](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: https://blog.csdn.net/qq_41357569/article/details/80572222

版权



[计组](#) 专栏收录该内容

6 篇文章 15 订阅

订阅专栏

实验要求:

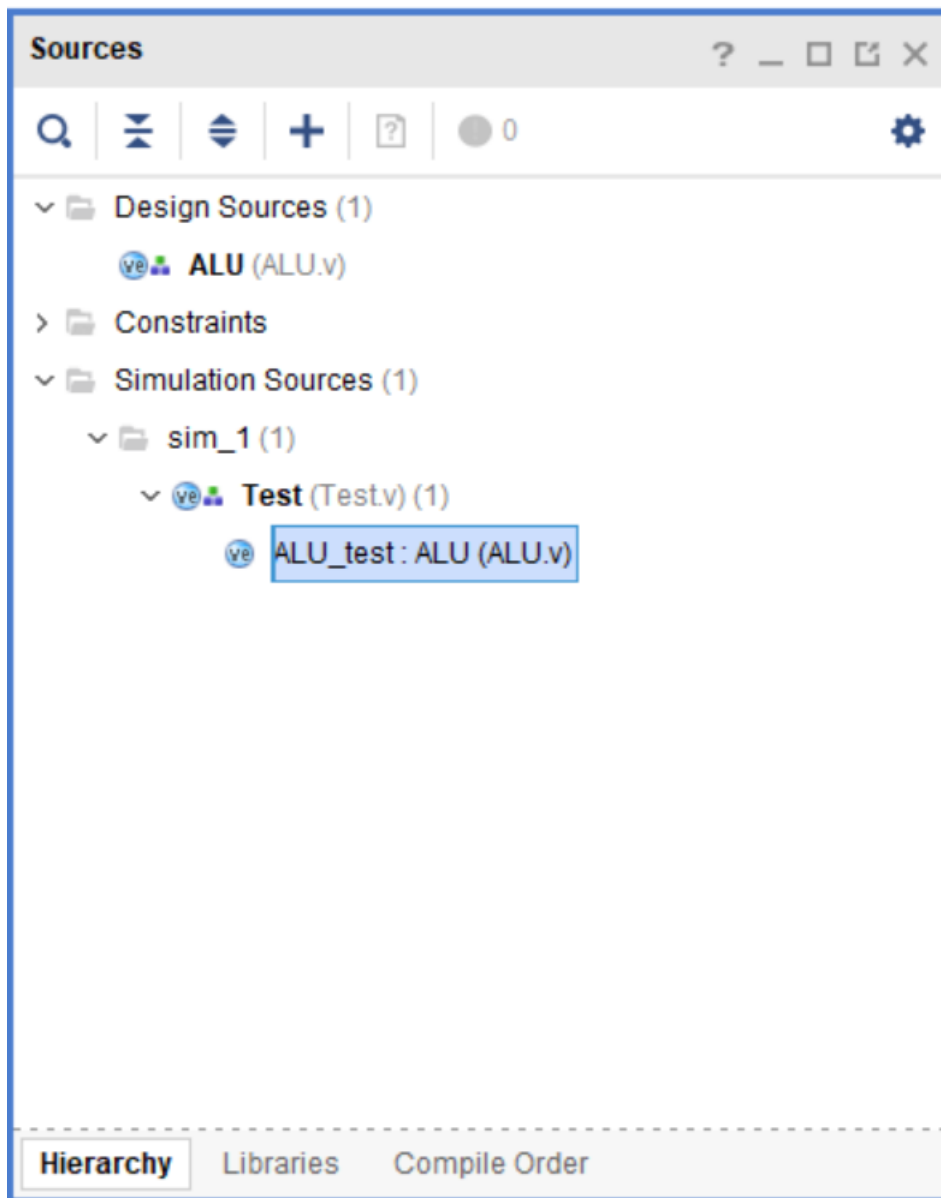
本实验要求设计一个具有8种运算功能的32位ALU, 并能够产生运算结果的表示: 结果为零标志ZF(Zero Flag)、进借位标志位CF()溢出标志OF(Overflow Flag)、符号位标志SF (Sign Flag) 和奇偶标志PF (Parity Flag), 标志位值解释见代码。ALU通过4根控制线ALU_OP[3:0]来选择其8种功能, 功能见表所示, 多余的8位用于后继运算的拓展备用。

功能表

ALU_OP[3:0]	ALU功能	功能说明
0000	and	按位与运算
0001	or	按位或运算
0010	xor	按位异或运算
0011	nor	按位或非运算
0100	add	算术加运算
0101	sub	算术减运算
0110	slt	若A<B, 则输出1;否则输出0
0111	sll	B逻辑左移A所指定的位数

实验IDE采用Vivado, 开发语言Verilog。

程序模块结构图:



ALU运算器模块代码：

```

`timescale 1ns / 1ps
//ALU模块
module ALU(OP,A,B,F,ZF,CF,OF,SF,PF);
    parameter SIZE = 32;//运算位数
    input [3:0] OP;//运算操作
    input [SIZE:1] A;//左运算数
    input [SIZE:1] B;//右运算数
    output [SIZE:1] F;//运算结果
    output ZF, //0标志位, 运算结果为0(全零)则置1, 否则置0
           CF, //进借位标志位, 取最高位进位C,加法时C=1则CF=1表示有进位,减法时C=0则CF=1表示有借位
           OF, //溢出标志位, 对有符号数运算有意义, 溢出则OF=1, 否则为0
           SF, //符号标志位, 与F的最高位相同
           PF; //奇偶标志位, F有奇数个1, 则PF=1, 否则为0
    reg [SIZE:1] F;
    reg C,ZF,CF,OF,SF,PF;//C为最高位进位
    always@(*)
    begin
        C=0;
        case(OP)
            4'b0000:begin F=A&B; end //按位与
            4'b0001:begin F=A|B; end //按位或
            4'b0010:begin F=A^B; end //按位异或
            4'b0011:begin F=~(A|B); end //按位或非
            4'b0100:begin {C,F}=A+B; end //加法
            4'b0101:begin {C,F}=A-B; end //减法
            4'b0110:begin F=A<B; end //A<B则F=1, 否则F=0
            4'b0111:begin F=B<<A; end //将B左移A位
        endcase
        ZF = F==0;//F全为0, 则ZF=1
        CF = C; //进位借位标志
        OF = A[SIZE]^B[SIZE]^F[SIZE]^C;//溢出标志公式
        SF = F[SIZE];//符号标志,取F的最高位
        PF = ~^F;//奇偶标志, F有奇数个1, 则F=1; 偶数个1, 则F=0
    end
endmodule

```

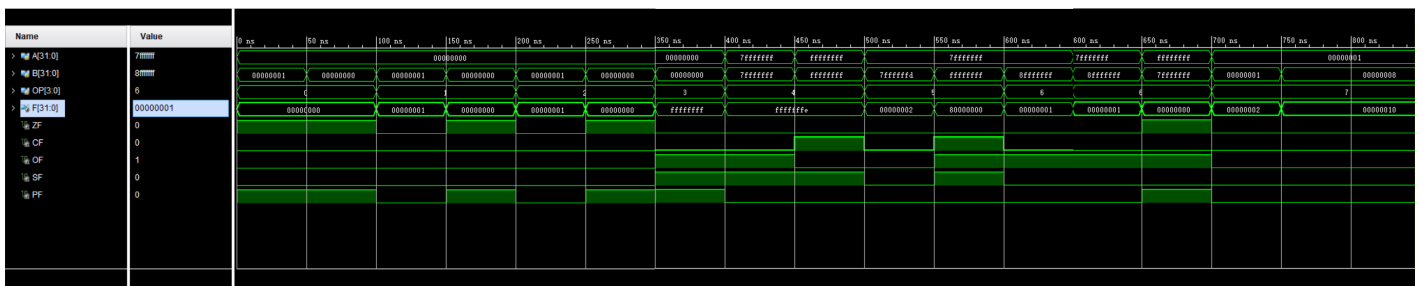
测试模块代码:

```

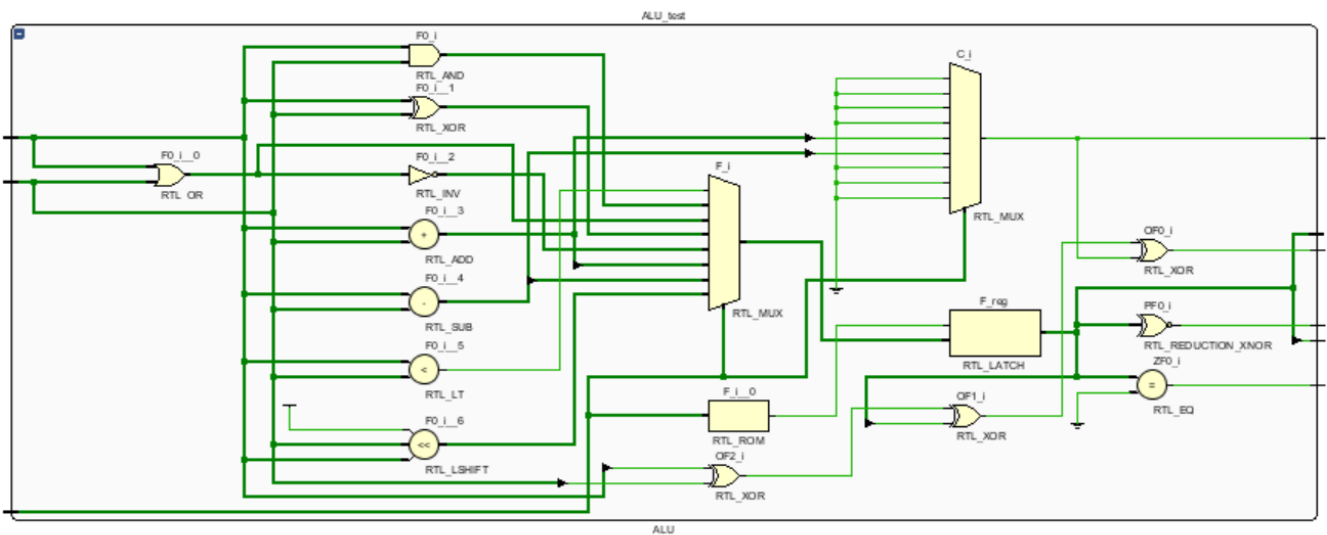
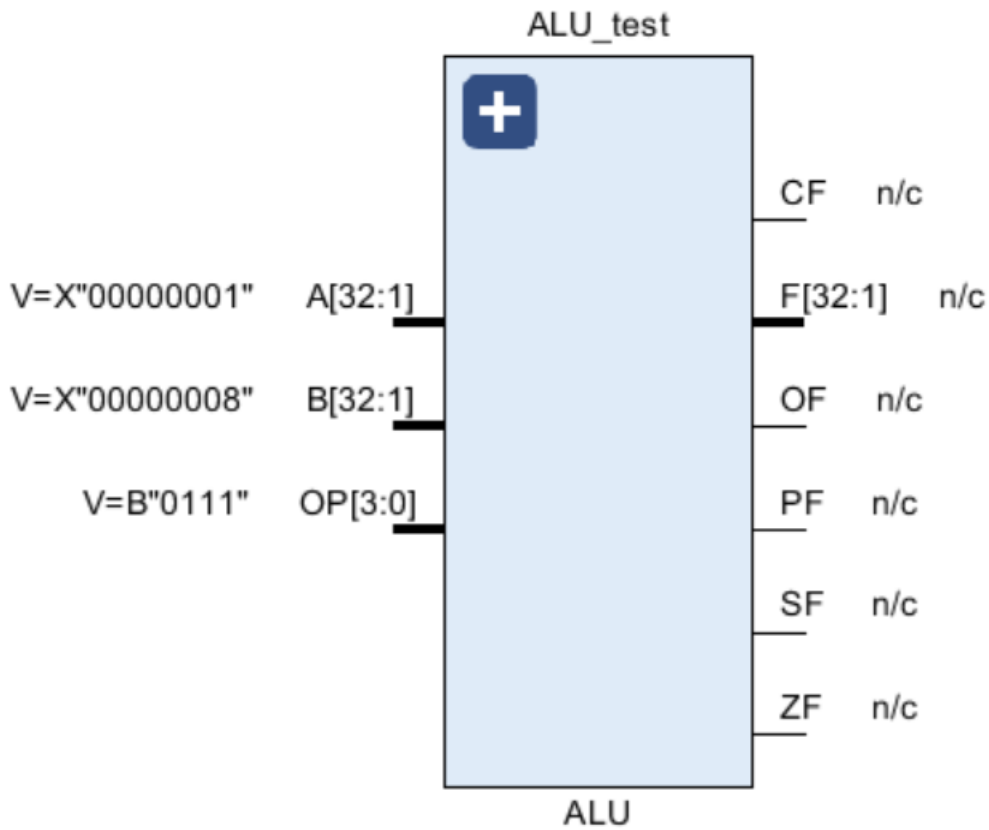
`timescale 1ns / 1ps
module Test();
    reg [31:0] A,B;
    reg [3:0] OP;
    initial//初始化, 每中运算擦拭两组数据
    begin
        /*按位与*/
        OP=4'b0000;A=32'h0000_0000; B=32'h0000_0001;#50;
        OP=4'b0000;A=32'h0000_0001; B=32'h0000_0001;#50;
        /*按位或*/
        OP=4'b0001;A=32'h0000_0000; B=32'h0000_0001;#50;
        OP=4'b0001;A=32'h0000_0000; B=32'h0000_0000;#50;
        /*按位异或*/
        OP=4'b0010;A=32'h0000_0000; B=32'h0000_0001;#50;
        OP=4'b0010;A=32'h0000_0000; B=32'h0000_0000;#50;
        /*按位或非*/
        OP=4'b0011;A=32'h0000_0000; B=32'h0000_0001;#50;
        OP=4'b0011;A=32'h0000_0000; B=32'h0000_0000;#50;
        /*算术加运算*/
        OP=4'b0100;A=32'h7FFF_FFFF; B=32'h7FFF_FFFF;#50;
        OP=4'b0100;A=32'hFFFF_FFFF; B=32'hFFFF_FFFF;#50;
        /*算术减运算*/
        OP=4'b0101;A=32'h7FFF_FFFF; B=32'h7FFF_FFFD;#50;
        OP=4'b0101;A=32'h7FFF_FFFF; B=32'hFFFF_FFFF;#50;
        /*A<B时,输出1, 否则输出0*/
        OP=4'b0110;A=32'h7FFF_FFFF; B=32'h8FFF_FFFF;#50;
        OP=4'b0110;A=32'hFFFF_FFFF; B=32'h7FFF_FFFF;#50;
        /*B逻辑左移A指定的位数*/
        OP=4'b0111;A=32'h0000_0001; B=32'h0000_0001;#50;
        OP=4'b0111;A=32'h0000_0001; B=32'h0000_0008;#50;
    end
    wire [31:0] F;
    wire ZF, CF, OF, SF, PF;
    ALU ALU_test(
        .OP(OP),
        .A(A),
        .B(B),
        .F(F),
        .ZF(ZF),
        .CF(CF),
        .OF(OF),
        .SF(SF),
        .PF(PF)
    );
endmodule

```

仿真波形图:



仿真电路图：



心得体会：

总的来说实验三不是很难吧，本来书上的要求是做OP[2:0]三位8中运算，标志位也没有那么多。章老师开了翻转课堂，给我们加了一些要求。这是本学期我们的第一个实验，很多Verilog的语法都忘记了，磕磕碰碰和组员讨论，debug。非常感谢另外两个小伙伴lym和tk。