

同济大学计算机系

数字逻辑课程实验报告



学 号 1652286

姓 名 李辉

专 业 计算机

授课老师 张冬冬

一、实验内容

在本次实验中，我们将使用 Verilog HDL 语言实现行为级 ALU 的设计和仿真。

二、模块建模

```
`timescale 1ns / 1ns
module alu(
    input [31:0] a,//32 位输入，操作数 1
    input [31:0] b,//32 位数，操作数 2
    input [3:0] aluc,//4 位输入，由 aluc 控制
    output reg[31:0] r,//32 位输出，由 a\b 经过 aluc 的操作生成
    output reg zero,//0 标志位
    output reg carry,//进位标志位
    output reg negative,//负数标志位
    output reg overflow//溢出标志位
);
parameter ADDU=4'b0000;
    parameter ADD=4'b0010;
    parameter SUBU=4'b0001;
    parameter SUB=4'b0011;
    parameter AND=4'b0100;
    parameter OR=4'b0101;
    parameter XOR=4'b0110;
    parameter NOR=4'b0111;
    parameter LUI1=4'b1000;
    parameter LUI2=4'b1001;
    parameter SLT=4'b1011;
    parameter SLTU=4'b1010;
    parameter SRA=4'b1100;
    parameter SLL=4'b1110;
    parameter SLR=4'b1111;
    parameter SRL=4'b1101;

    reg [32:0] mm;
    reg[31:0] cc;

    always@(*)begin
        case(aluc)
            ADDU: begin
                r=a+b;
                mm=a+b;
                if(!r)
                    zero=1;
                else
                    zero=0;
                if(mm[32]==1)
                    carry=1;
                else
                    carry=0;
                negative=r[31];
                overflow=0;
            end
            SUBU: begin
                r=a-b;
                if(!r)
                    zero=1;
                else
                    zero=0;
                if(a<b)
                    carry=1;
                else
                    carry=0;
                negative=r[31];
                overflow=0;
            end
            ADD: begin
                r=a+b;
                if(!r)
                    zero=1;
                else
                    zero=0;
                carry=0;
                negative=r[31];
            end
            if(a[31]==1&&b[31]==1&&r[31]==0)
                overflow=1;
            else
            if(a[31]==0&&b[31]==0&&r[31]==1)
                overflow=1;
            else
                overflow=0;
            end
            SUB: begin
                r=a-b;
                if(!r)
                    zero=1;
                else
                    zero=0;
            end
        endcase
    end
```

```

        carry=0;
        negative=r[31];

if((a[31]==1&&b[31]==0&&r[31]==0)||((a[31]==0)&&b[31]
]==1&&r[31]==1))

        overflow=1;
        else
        overflow=0;
end
SRA: begin
    if(a>0)
        carry=b[a-1];
    else
        carry=0;
        r=b>>>a;
        if(!r)
            zero=1;
        else
            zero=0;
            negative=r[31];
            overflow=0;
        end
    end
SRL: begin
    if(a>0)
        carry=b[a-1];
    else
        carry=0;
        r=b>>a;
        if(!r)
            zero=1;
        else
            zero=0;
            negative=r[31];
            overflow=0;
        end
    end
SLL: begin
    r=b<<a;
    if(!r)
        r=1;
    else
        r=0;
        negative=r[31];
        overflow=0;
        if(a>0)
            carry=b[32-a];
        else
            carry=0;
        end
    end
AND: begin

```

```

        r=a&b;
        if(!r)
            zero=1;
        else
            zero=0;
            carry=0;
            negative=r[31];
            overflow=0;
        end
    end
OR: begin
    r=a|b;
    if(!r)
        zero=1;
    else
        zero=0;
        carry=0;
        negative=r[31];
        overflow=0;
    end
XOR: begin
    r=a^b;
    if(!r)
        zero=1;
    else
        zero=0;
        carry=0;
        negative=r[31];
        overflow=0;
    end
    end
NOR: begin
    r=~(a|b);
    if(!r)
        zero=1;
    else
        zero=0;
        carry=0;
        negative=r[31];
        overflow=0;
    end
    end
SLTU: begin
    r=a<b?1:0;
    if((a-b)==0)
        zero=1;
    else
        zero=0;
        if(a<b)
            carry=1;
        else
            carry=0;
        end
    end

```

```

        negative=r[31];
        overflow=0;
    end
    SLT: begin
        cc=a-b;
        if(cc[31]==1)
            r=1;
        else
            r=0;
        if((a-b)==0)
            zero=1;
        else
            zero=0;
            carry=0;
            if(cc[31]==1)
                negative=1;
            else
                negative=0;
                overflow=0;
            end
        LUI1,LUI2: begin
            r={b[15:0],16'b0};
            if(!r)
                zero=1;
            else
                zero=0;
                carry=0;
                negative=r[31];
                overflow=0;
            end
            default:
                r=a+b;
        endcase
    end
endmodule

```

三、测试模块建模

```

`timescale 1ns / 1ps
module alu_tb;
    reg [31:0]a;
    reg [31:0]b;
    reg [3:0]aluc;
    wire [31:0]r;
    wire zero;
    wire carry;
    wire negative;
    wire overflow;

```

```

    alu uut (.a(a),
        .b(b),
        .aluc(aluc),
        .r(r),
        .zero(zero),
        .carry(carry),
        .negative(negative),
        .overflow(overflow));

    initial
    begin
        aluc=4'b0000;
        a=0;
        b=0;
        #10
        a=0;
        b=1;
        #10
        a={32{1'b1}};
        b=1;
        #10
        a={32{1'b1}};
        b={32{1'b1}};
        #10
        a={1'b0,{31{1'b1}}};
        b={1'b0,{31{1'b1}}};
        #50

        aluc=4'b0010;
        a=0;
        b=0;
        #10
        a=0;
        b=1;
        #10
        a={32{1'b1}};
        b=1;
        #10
        a={32{1'b1}};
        b={32{1'b1}};
        #10
        a={1'b0,{31{1'b1}}};
        b={1'b0,{31{1'b1}}};
        #50

        aluc=4'b0001;
        a=0;
        b=0;
        #10

```

```

a=0;
b=1;
#10
a={32{1'b1}};
b=1;
#10
a={32{1'b1}};
b={32{1'b1}};
#10
a={1'b0,{31{1'b1}}};
b={1'b0,{31{1'b1}}};
#50

aluc=4'b0011;
a=0;
b=0;
#10
a=0;
b=1;
#10
a={32{1'b1}};
b=1;
#10
a={32{1'b1}};
b={32{1'b1}};
#10
a={1'b0,{31{1'b1}}};
b={1'b0,{31{1'b1}}};
#10
a={1'b0,{31{1'b1}}};
b={1'b1,{31{1'b1}}};

aluc=4'b0100;
a={{8{1'b1}}, {8{1'b0}}, {8{1'b0}}, {8{1'b1}}};
b={{8{1'b1}}, {8{1'b1}}, {8{1'b0}}, {8{1'b0}}};
#10
aluc=4'b0101;
a={{8{1'b1}}, {8{1'b0}}, {8{1'b0}}, {8{1'b1}}};
b={{8{1'b1}}, {8{1'b1}}, {8{1'b0}}, {8{1'b0}}};
#10
aluc=4'b0110;
a={{8{1'b1}}, {8{1'b0}}, {8{1'b0}}, {8{1'b1}}};
b={{8{1'b1}}, {8{1'b1}}, {8{1'b0}}, {8{1'b0}}};
#10

```

```

aluc=4'b0111;
a={{8{1'b1}}, {8{1'b0}}, {8{1'b0}}, {8{1'b1}}};
b={{8{1'b1}}, {8{1'b1}}, {8{1'b0}}, {8{1'b0}}};

aluc =4'b1000;
b={{16{1'b0}}, {16{1'b1}}};
#10
aluc =4'b1001;
b={{16{1'b1}}, {16{1'b0}}};

aluc = 4'b1011;
a={1'b1,{31{1'b0}}};
b={1'b0,{31{1'b1}}};
#10
aluc = 4'b1010;

aluc=4'b1100;
b={1'b0,{31{1'b1}}};
a=32'd32;
#10
b={1'b1,{31{1'b0}}};
a=32'd32;
#10
b=32'b101010;
a=32'd3;
aluc=4'b1101;
b={1'b0,{31{1'b1}}};
a=32'd32;
#10
b={1'b1,{31{1'b0}}};
a=32'd32;
#10
b=32'b101010;
a=32'd3;

aluc=4'b1111;
b={16'b0,{16{1'b1}}};
a=32'd16;
#10
b={16'b1,{16{1'b0}}};
a=32'd16;

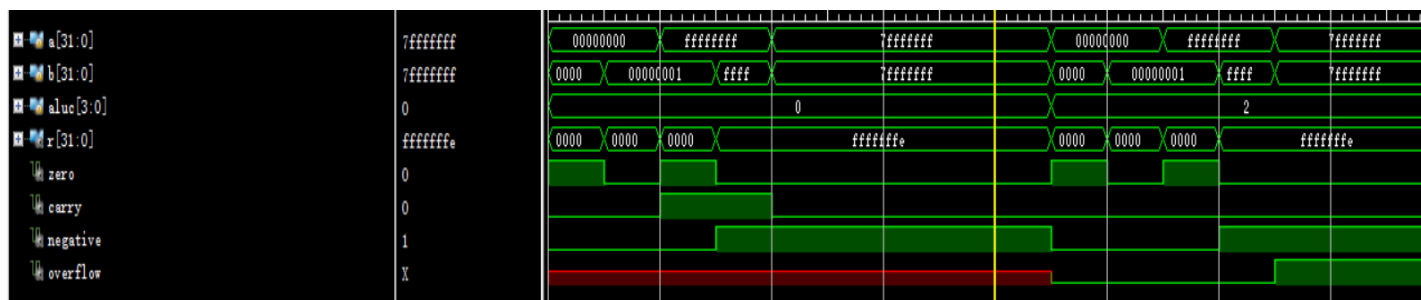
end
endmodule

```

四、实验结果

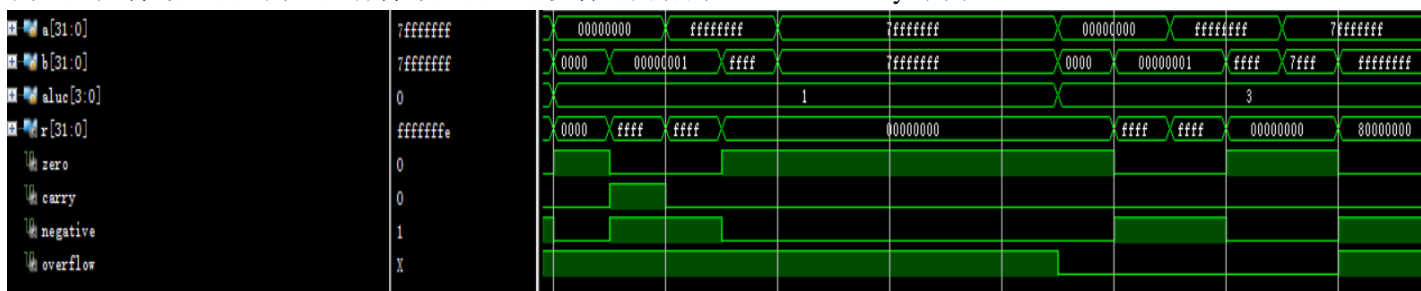
仿真波形分析：

①



aluc 为 0 是无符号加法，为 2 是有符号加法。可以看出两者的 overflow, carry 不同。

②



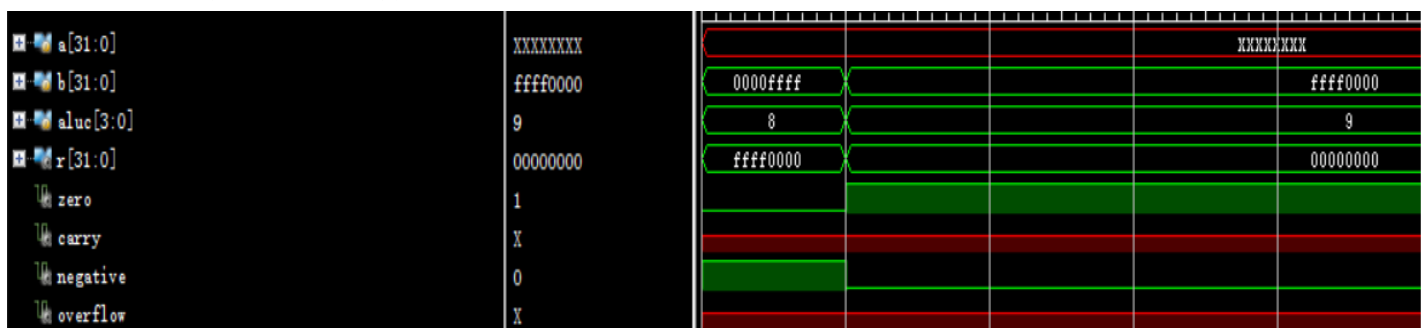
aluc 为 1 是无符号减法，3 是有符号减法。结果正确。

③



分别进行了与、或、异或、或非运算，结果正确。

④



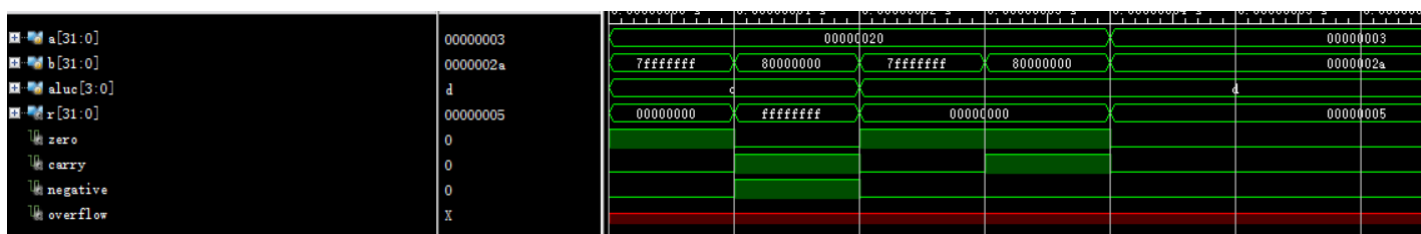
进行了置高位立即数运算，结果正确。

⑤



进行了有符号数和无符号数的比较，r 正确，其余标志位也正确。

⑥



⑦

左移，无论是算术还是逻辑，全都在最低位补 0。