一、实验内容

在本次实验中，将使用 Verilog HDL 语言设计实现：

(1) 行为级 ALU 的设计和仿真

二、硬件逻辑图(无)

三、模块建模

功能描述：能够实现有符号数加减、无符号数加减、与、或、异或、或非、置最高位立即数、有符号数大小比较、无符号数大小比较、算术右移、算术左移、逻辑右移和逻辑左移等功能。

```
module alu(
        input [31:0] a,
        input [31:0] b,
        input [3:0] aluc,
        output reg [31:0] r,
        output reg zero,
        output reg carry,
        output reg negative,
        output reg overflow
        );
          always @(*) begin
                    case (aluc)
                            4'b0000: begin        //addu
                                    r= a + b;
                                    if(r==0)
                                        zero=1;
                                    else
                                        zero=0;
                                    if(r[31]==1)
                                        negative=1;
                                    else
                                        negative=0;
                                    if(r<a||r<b)
                                        carry=1;
                                    else
                                        carry=0;
                            end
```

```
4'b0001: begin
     r= a - b;
     if(r==0)
        zero=1;
     else
        zero=0;
     if(r[31]==1)
        negative=1;
     else
        negative=0;
     if(a<b)
        carry=1;
     else
        carry=0;
end
4'b0010: begin
       r= $signed(a) + $signed(b);
       if(r==0)
          zero=1;
       else
          zero=0;

       if(r[31]==0)
          negative=0;
       else
          negative=1;

       if(a[31]==0&&b[31]==0&&r[31]==1)
           overflow=1;
       else if(a[31]==1&&b[31]==1&&r[31]==0)
          overflow=1;
       else
          overflow=0;
end
4'b0011:begin //sub    3
     r= $signed(a) - $signed(b);
     if(r==0)
       zero=1;
     else
       zero=0;

     if(r[31]==1)
       negative=1;
     else
```

```
       negative=0;

     if(a[31]==0&&b[31]==1&&r[31]==1)
       overflow=1;
     else if(a[31]==1&&b[31]==0&&r[31]==0)
       overflow=1;
     else
       overflow=0;
end
4'b0100:begin// and
     r=a&b;
     if(r==0)
       zero=1;
     else
       zero=0;

     if(r[31]==1)
       negative=1;
     else
       negative=0;
end
4'b0101:begin//or
     r=a|b;
     if(r==0)
       zero=1;
     else
       zero=0;
    if(r[31]==1)
       negative=1;
    else
       negative=0;
end
4'b0110:begin //xor
     r=a^b;
     if(r==0)
        zero=1;
     else
        zero=0;

     if(r[31]==1)
        negative=1;
      else
        negative=0;
end
```

```
4'b0111:begin //nor
        r=~(a|b);
        if(r==0)
            zero=1;
        else
            zero=0;

        if(r[31]==1)
            negative=1;
        else
            negative=0;
end
4'b1000:begin //lui
        r={b[15:0],16'b0};
        if(r==0)
          zero=1;
        else
          zero=0;

        if(r[31]==1)
            negative=1;
        else
          negative=0;
end
4'b1001:begin//lui
        r={b[15:0],16'b0};
        if(r==0)
            zero=1;
        else
            zero=0;

        if(r[31]==1)
            negative=1;
        else
            negative=0;
end
4'b1010:begin//sltu
        r=(a<b)?1:0;
        if(a-b==0)
            zero=1;
        else
            zero=0;

        if(a<b)
```

```
                carry=1;
            else
                carry=0;

            if(r[31]==1)
                  negative=1;
            else
                  negative=0;
end
4'b1011:begin //slt
            r=($signed(a) < $signed(b))?1:0;
            if($signed(a) - $signed(b)==0)
                  zero=1;
            else
                  zero=0;

            if($signed(a) - $signed(b)<0)
                negative=1;
            else
              negative=0;

end
4'b1100:begin//sra--
            r=$signed(b)>>>a;

            if(a<=32&&a>=1)
                  carry=b[a-1];
            else if(a>32)
                  carry=b[31]; //          λ
            else
                  carry=0;



            if(r==0)
                zero=1;
            else
                zero=0;

          if(r[31]==1)
                negative=1;
          else
                negative=0;
end
```

```
4'b1101:begin//srl
      r=b>>a;
      if(r==0)
        zero=1;
      else
        zero=0;

      if(r[31]==1)
         negative=1;
      else
         negative=0;

      if(a<=32&&a>=1)
         carry=b[a-1];
       else
         carry=0;
end
4'b1110:begin//sll/sla
      r=b<<a;
      if(r==0)
        zero=1;
      else
        zero=0;

      if(r[31]==1)
         negative=1;
      else
         negative=0;

      if(a<=32&&a>=1)
         carry=b[32-a];
      else
         carry=0;
end
4'b1111:begin//sll/sla
      r=b<<a;
      if(r==0)
        zero=1;
      else
        zero=0;

      if(r[31]==1)
         negative=1;
      else
```

```
                                negative=0;

                            if(a<=32&&a>=1)
                                carry=b[32-a];
                           else
                                carry=0;
                    end
                      default: begin
                           r = 32'b0;
                      end
                endcase
          end


endmodule
```

四、测试模块建模

```
module alu_tb;
reg [31:0] a, b;
reg [3:0] aluc;
wire [31:0] r;
wire zero, carry, negative, overflow;

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

initial begin
        a = 0; b = 0; aluc = 0;
        #10;
        //无符号加法
        a = 32'h00000001; b = 32'h00000001; aluc = 4'b0000; //不溢出，结果为32h000000002
        #10;
        a = 32'h11111111; b = 32'h0000000F; aluc = 4'b0000; //溢出，结果为 32h000000000
        #10;
```

```
//  无符号减法
    a = 32'h00000002; b = 32'h00000001; aluc = 4'b0001;//不溢出，结果为 32'h00000001
    #10;
    a = 32'h00000002; b = 32'h00000011; aluc = 4'b0001;//溢出，结果为 FFFFFFF1
    #10;
    //有符号加法
    a = 32'h00000001; b = 32'h00000001; aluc = 4'b0010; // 不溢出，结果为 32h000000002
    #10;
    a = 32'h01111111; b = 32'h0000000F; aluc = 4'b0010; //溢出，结果为 32h1000000
    #10;
    a = 32'h11111111; b = 32'h1000000F; aluc = 4'b0010; //溢出
    #10;
    //有符号减法
    a = 32'h00000002; b = 32'h00000001; aluc = 4'b0011; //不溢出
    #10;
    a = 32'h80000000; b = 32'h0000000F; aluc = 4'b0011; //溢出
    #10;
   //与 0100
    a = 32'h80000000; b = 32'h00000001; aluc = 4'b0100;
     #10;
   //或 0101
    a = 32'h80000000; b = 32'h00000001; aluc = 4'b0101;
     #10;
   //异或 0110
a = 32'h80000000; b = 32'h00000001; aluc = 4'b0110;
    #10;
   //或非 0111
   a = 32'h80000000; b = 32'h00000001; aluc = 4'b0111;
   #10;
   //置高位立即数 1001/1000
   a = 32'h80000000; b = 32'h00000001; aluc = 4'b1000;
   #10;
   a = 32'h80000000; b = 32'h00000001; aluc = 4'b1001;
   #10;
  //有符号数比较 1011
   a = 32'h00000001; b = 32'hffffffff; aluc = 4'b1011;
   #10;
   a = 32'hffffffff; b = 32'h0000ffff; aluc = 4'b1011;
   #10;
   //无符号数比较 1010
   a = 32'h80000000; b = 32'h00000001; aluc = 4'b1010;
   #10;
   //算术右移 1100
```

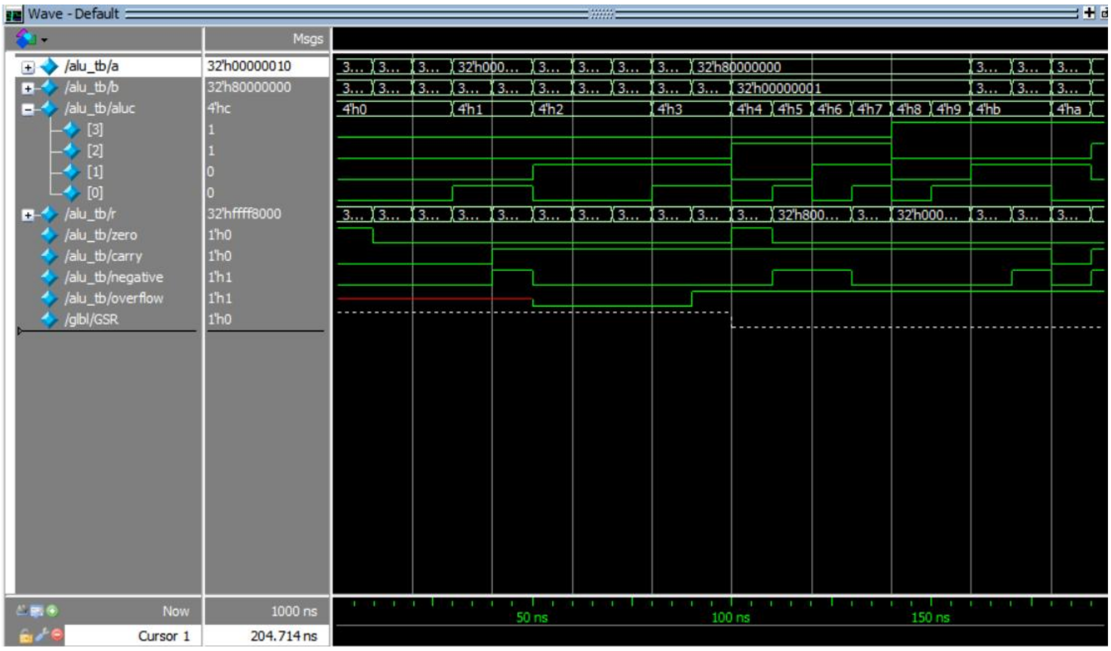
```
    a = 32'h00000008; b = 32'hffffffff; aluc = 4'b1100;
    #10;
    a = 32'h00000010; b = 32'h80000000; aluc = 4'b1100;
    #10;
    //逻辑左移/算术左移 1111/1110
    a = 32'h80000000; b = 32'h00000001; aluc = 4'b1111;
    #10;
    a = 32'h00000020; b = 32'h00100001; aluc = 4'b1110;
    #10;
    //逻辑右移 1101
    a = 32'h80000000; b = 32'h00000001; aluc = 4'b1101;
    #10;
    a = 32'h00000020; b = 32'h00100001; aluc = 4'b1101;
    #10;

end

endmodule
```

五、实验结果

Modelsim 仿真波形图



Wave - Default

```
                 Msgs
/alu_tb/a        32'h00000010
/alu_tb/b        32'h80000000
/alu_tb/aluc     4'hc
  [3]            1
  [2]            1
  [1]            0
  [0]            0
/alu_tb/r        32'hffff8000
/alu_tb/zero     1'h0
/alu_tb/carry    1'h0
/alu_tb/negative 1'h1
/alu_tb/overflow 1'h1
/glbl/GSR        1'h0

Now      1000 ns
Cursor 1 204.714 ns
```

（PS：overflow 红色那一段不是出错，而是由于一开始进行无符号的加减，对 overflow 位的值没有要求，故没有设置，显示为 x（未知）的状态）