同济大学计算机系

数字逻辑课程实验报告

学　　号 2252707

姓　　名 陈艺天

专　　业 计算机科学与技术

授课老师 张冬冬

1. 行为级 ALU

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

<ALU>模块建模

```
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
    );
    reg [32:0] tmp;

    always @(*) begin

        casex(aluc)

            4'b0000:/*Addu*/begin
                {carry, r} = a + b;
                negative = r[31];
                zero = !r;
            end

            4'b0010:/*Add*/begin
                r = a + b;
                negative = r[31];
                overflow =
                (a[31]&b[31]&~r[31])|(~a[31]&~b[31]&r[31]);
                zero = !r;
            end
            4'b0001:/*Subu*/begin
                r = a - b;
                negative = r[31];
```

```
               tmp = 33'b1_00000000_00000000_00000000_00000000 + a - b;
               carry = ~tmp[32];
               zero = !r;
           end
           4'b0011:/*Sub*/begin
               r = a - b;
               negative = r[31];
               overflow =
               (~a[31]&b[31]&r[31])|(a[31]&~b[31]&~r[31]);
               zero = !r;
           end
           4'b0100:/*And*/begin
               r = a & b;
               negative = r[31];
               zero = !r;
           end
           4'b0101:/*Or*/begin
               r = a | b;
               negative = r[31];
               zero = !r;
           end
           4'b0110:/*Xor*/begin
               r = a ^ b;
               negative = r[31];
               zero = !r;
           end
           4'b0111:/*Nor*/begin
               r = ~(a | b);
               negative = r[31];
               zero = !r;
           end
           4'b100x:/*Lui*/begin
               r = {b[15:0],16'b0};
               negative = r[31];
               zero = !r;
           end
           4'b1011:/*Slt*/begin
               if(a[31]==b[31])
                   r = (a < b) ? 1: 0;
               else
                   r = a[31];
               negative = r[0];
               zero = (a==b)?1:0;
```

```
            end
            4'b1010:/*Sltu*/begin
                r = (a < b) ? 1: 0;
                negative = r[31];
                carry = (a < b) ? 1: 0;
                zero = (a==b)?1:0;
            end
            4'b1100:/*Sra*/begin
                if(!b[31])
                    r=(b>>(a-1));
                else
                    r=(b>>(a-1))|~(32'b11111111111111111111111111111
111>>(a-1));
                carry = r[0];
                if(!b[31])
                    r = r>>1;
                else begin
                    r = r>>1;
                    r = {1'b1,r[30:0]};
                end
                negative = r[31];
                zero = !r;
            end
            4'b111x:/*Sll/sla*/begin
                r = b << (a-1);
                carry = r[31];
                r = r << 1;
                negative = r[31];
                zero = !r;
            end
            4'b1101:/*srl*/begin
                r = b >> (a-1);
                carry = r[0];
                r = r>>1;
                negative = r[31];
                zero = !r;
            end
            default:
                ;
        endcase

    end

endmodule
```

三、测试模块建模

<加法模块测试>

```
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 inst(a,b,aluc,r,zero,carry,negative,overflow);


    //test addu
    initial begin
        aluc = 4'b0000;
        a = 32'b1000_0010_0000;
        b = 32'b1000_0010_0000;

        #40
        aluc = 4'b0000;
        a = 32'b0000_0000_0000;
        b = 32'b0000_0000_0000;

        #40
        aluc = 4'b0000;
        a = 32'b1000_0010_0000_0000_0000_0000_0000_0010;
        b = 32'b1000_0010_0000_0001_0000_0000_0000_0000;

        #40
        aluc = 4'b0000;
        a = 32'b0011_0110_0000_0000_0000_0000_0000_0010;
        b = 32'b1000_0010_0000_0001_0000_0000_0000_0000;
    end


endmodule
```

```
module alu_tb_add;

    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 inst(a,b,aluc,r,zero,carry,negative,overflow);


    //test addu
    initial begin
        aluc = 4'b0010;
        a = 32'b1000_0010_0000;
        b = 32'b1000_0010_0000;

        #40
        aluc = 4'b0010;
        a = 32'b0000_0000_0000;
        b = 32'b0000_0000_0000;

        #40
        aluc = 4'b0010;
        a = 32'b1000_0010_0000_0000_0000_0000_0000_0010;
        b = 32'b1000_0010_0000_0001_0000_0000_0000_0000;

        #40
        aluc = 4'b0010;
        a = 32'b0011_0110_0000_0000_0000_0000_0000_0010;
        b = 32'b1000_0010_0000_0001_0000_0000_0000_0000;
    end


endmodule
```

<减法模块测试>

```
module alu_tb_sub;

    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 inst(a,b,aluc,r,zero,carry,negative,overflow);


    //test addu
    initial begin
        aluc = 4'b0011;
        a = 32'b1000_0010_0000;
        b = 32'b1000_0010_0000;

        #40
        aluc = 4'b0011;
        a = 32'b0000_0000_0000;
        b = 32'b1000_0000_0000;

        #40
        aluc = 4'b0011;
        a = 32'b1000_0010_1111_0000_1111_0000_0000_0010;
        b = 32'b1000_0010_0000_0001_0000_0000_0000_0000;

        #40
        aluc = 4'b0011;
        a = 32'b0011_0110_0000_0000_0000_0000_0000_0010;
        b = 32'b1000_0010_0000_0001_0000_0000_0000_0000;

        #40
        aluc = 4'b0011;
        a = 32'b1111_0110_0000_0000_0000_0000_0000_0010;
        b = 32'b0000_0010_0000_0001_0000_0000_0000_0000;
    end


endmodule
```

```
module alu_tb_subu;

    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 inst(a,b,aluc,r,zero,carry,negative,overflow);


    //test addu
    initial begin

        aluc = 4'b0011;
        a = 32'b1000_0010_0000;
        b = 32'b1000_0010_0000;

        #40
        aluc = 4'b0011;
        a = 32'b0000_0000_0000;
        b = 32'b1000_0000_0000;

        #40
        aluc = 4'b0011;
        a = 32'b1000_0010_1111_0000_1111_0000_0000_0010;
        b = 32'b1000_0010_0000_0001_0000_0000_0000_0000;

        #40
        aluc = 4'b0011;
        a = 32'b0011_0110_0000_0000_0000_0000_0000_0010;
        b = 32'b1000_0010_0000_0001_0000_0000_0000_0000;
    end


endmodule
```

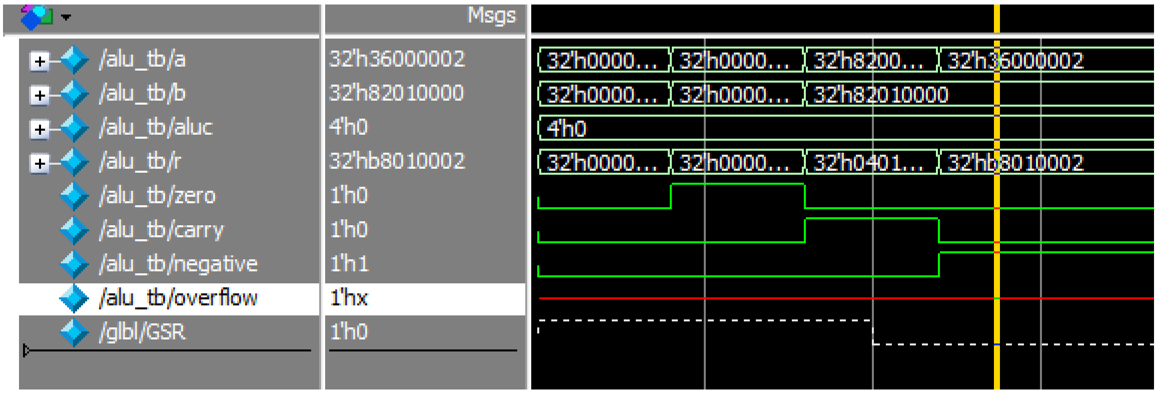
<位运算模块测试>

```
module alu_tb_shift;

    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 inst(a,b,aluc,r,zero,carry,negative,overflow);


    //test addu
    initial begin
        aluc = 4'b1011;
        a = 32'b1000_0010_0000;
        b = 32'b1000_0010_0000;

        #40
        aluc = 4'b1010;
        a = 32'b0000_0000_0000;
        b = 32'b1000_0000_0000;

        #40
        aluc = 4'b1100;
        b = 32'b1011_0110_0000_0000_0000_0000_0000_0010;
        a = 32'b0000_0000_0000_0000_0000_0000_0000_1000;

        #40
        aluc = 4'b1111;
        b = 32'b1011_0110_0000_0000_0000_0000_0000_0010;
        a = 32'b0000_0000_0000_0000_0000_0000_0000_1000;

        #40
        aluc = 4'b1101;
        a = 32'b1111_0110_0000_0000_0000_0000_0000_0010;
        b = 32'b0000_0010_0000_0001_1111_1111_0000_0000;
    end


endmodule
```

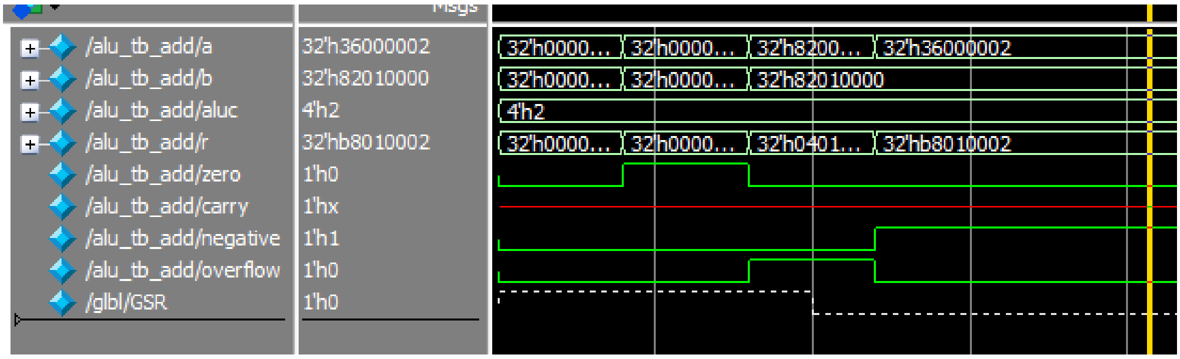
四、实验结果

Modelsim 仿真图

<1>加法模块

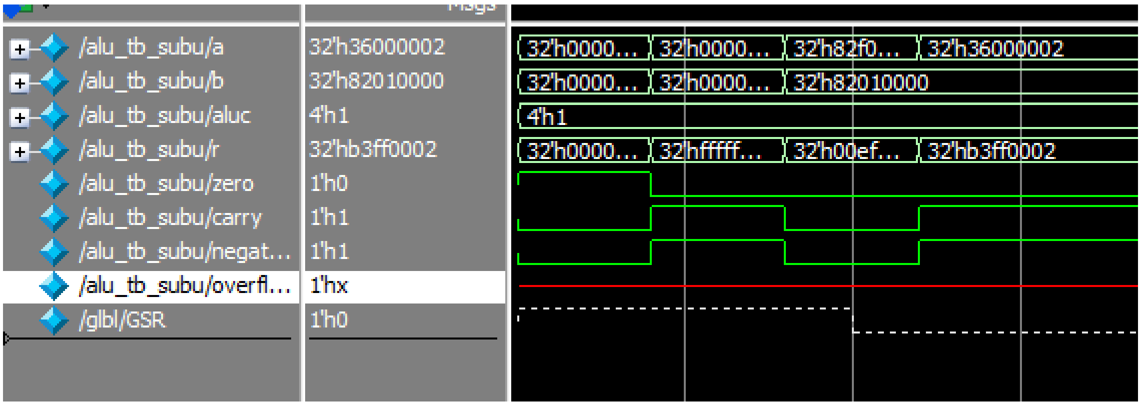


	Msgs
/alu_tb/a	32'h36000002
/alu_tb/b	32'h82010000
/alu_tb/aluc	4'h0
/alu_tb/r	32'hb8010002
/alu_tb/zero	1'h0
/alu_tb/carry	1'h0
/alu_tb/negative	1'h1
/alu_tb/overflow	1'hx
/glbl/GSR	1'h0

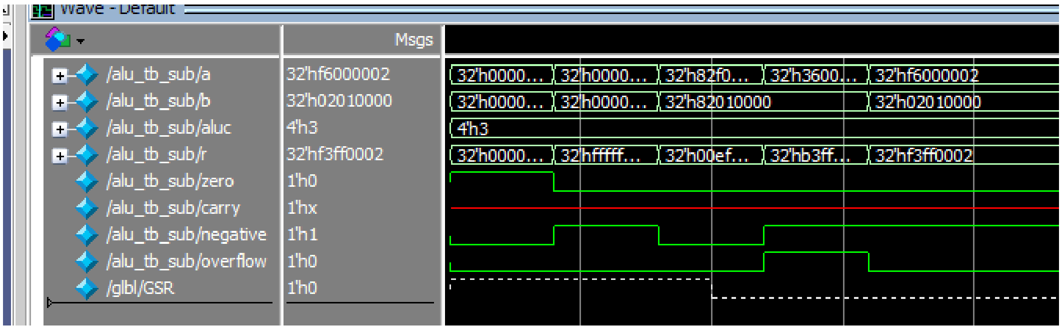


	Msgs
/alu_tb_add/a	32'h36000002
/alu_tb_add/b	32'h82010000
/alu_tb_add/aluc	4'h2
/alu_tb_add/r	32'hb8010002
/alu_tb_add/zero	1'h0
/alu_tb_add/carry	1'hx
/alu_tb_add/negative	1'h1
/alu_tb_add/overflow	1'h0
/glbl/GSR	1'h0

<2>减法模块

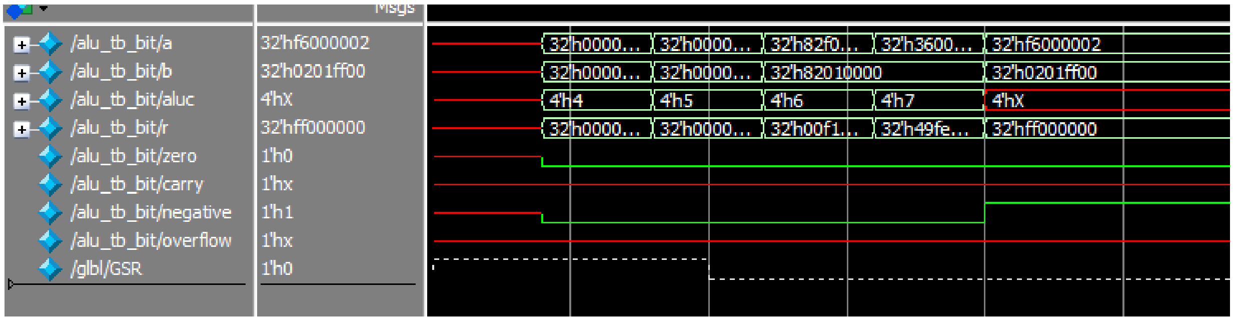


	Msgs
/alu_tb_subu/a	32'h36000002
/alu_tb_subu/b	32'h82010000
/alu_tb_subu/aluc	4'h1
/alu_tb_subu/r	32'hb3ff0002
/alu_tb_subu/zero	1'h0
/alu_tb_subu/carry	1'h1
/alu_tb_subu/negat...	1'h1
/alu_tb_subu/overfl...	1'hx
/glbl/GSR	1'h0

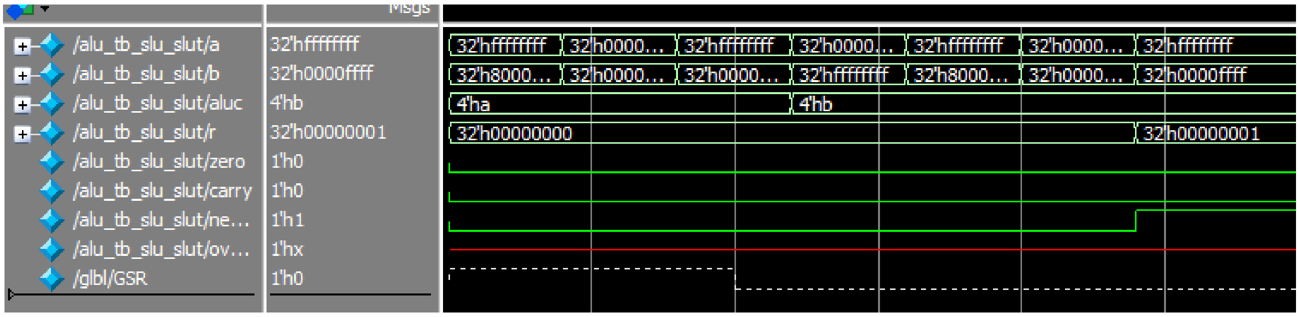


	Msgs
/alu_tb_sub/a	32'hf6000002
/alu_tb_sub/b	32'h02010000
/alu_tb_sub/aluc	4'h3
/alu_tb_sub/r	32'hf3ff0002
/alu_tb_sub/zero	1'h0
/alu_tb_sub/carry	1'hx
/alu_tb_sub/negative	1'h1
/alu_tb_sub/overflow	1'h0
/glbl/GSR	1'h0

<位运算模块>



	Msgs
/alu_tb_bit/a	32'hf6000002
/alu_tb_bit/b	32'h0201ff00
/alu_tb_bit/aluc	4'hX
/alu_tb_bit/r	32'hff000000
/alu_tb_bit/zero	1'h0
/alu_tb_bit/carry	1'hx
/alu_tb_bit/negative	1'h1
/alu_tb_bit/overflow	1'hx
/glbl/GSR	1'h0



	Msgs
/alu_tb_slu_slut/a	32'hffffffff
/alu_tb_slu_slut/b	32'h0000ffff
/alu_tb_slu_slut/aluc	4'hb
/alu_tb_slu_slut/r	32'h00000001
/alu_tb_slu_slut/zero	1'h0
/alu_tb_slu_slut/carry	1'h0
/alu_tb_slu_slut/ne...	1'h1
/alu_tb_slu_slut/ov...	1'hx
/glbl/GSR	1'h0