计算机组成原理实验报告
LAB01

题目：运算器部件 ALU

姓名：林祥

学号：PB16020923

实验目的

1、复习 Verilog 语法

2、知道如何用 Verilog 实现运算器部件 ALU

实验内容

1、设计一个算术运算单元 ALU

2、采用纯组合逻辑设计

3、32bit 宽

4、完成 7 种运算功能

实验分析

1、模块化设计，alu 一个模块，计算斐波那契数列一个 top 模块。

2、alu 模块使用 case 语句判断 7 种操作类型。

实验结果

（1）ALU 测试

输入运算数

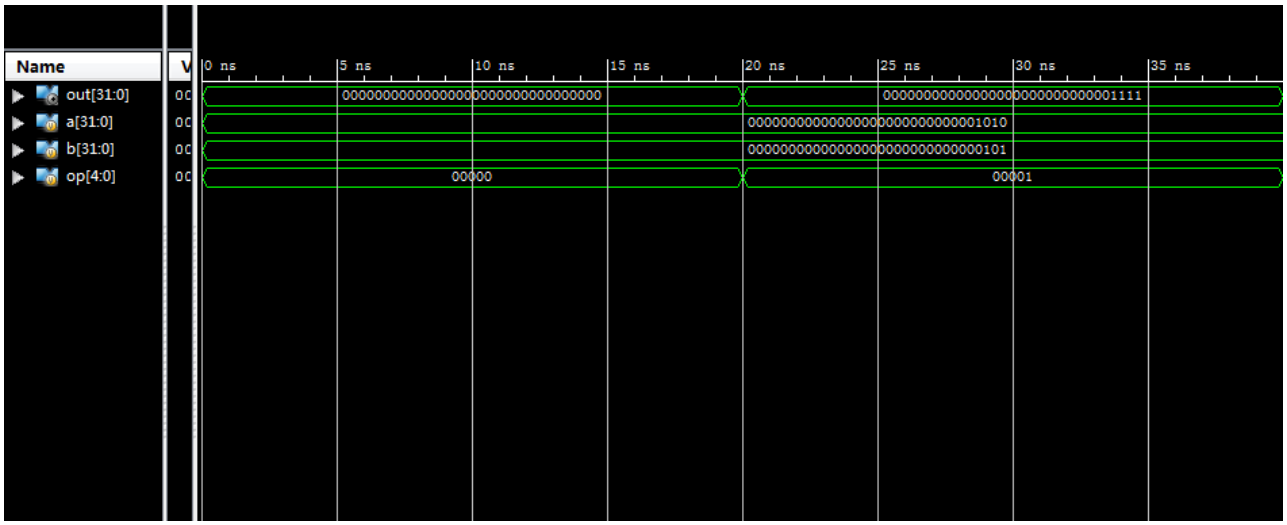
a=10(=32’b00000000000000000000000000001010)

b=5 (=32’b00000000000000000000000000000101)

输入操作码

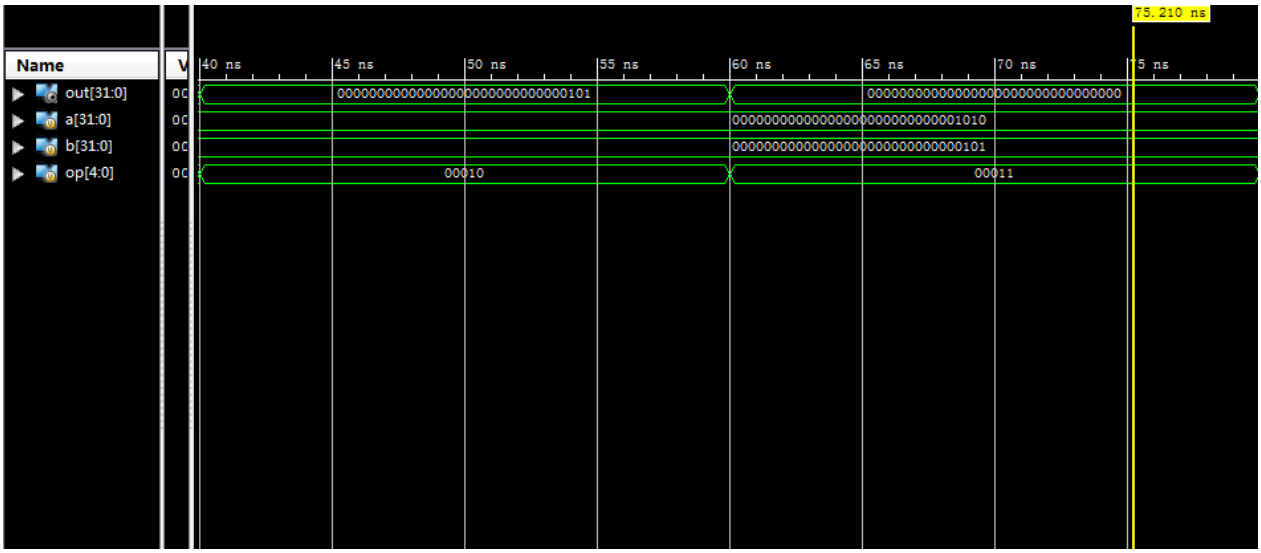
op=0 时， out=0

op=1 时， out=15(=32’b00000000000000000000000000001111)



op=2 时， out=5(=32’b00000000000000000000000000000101)

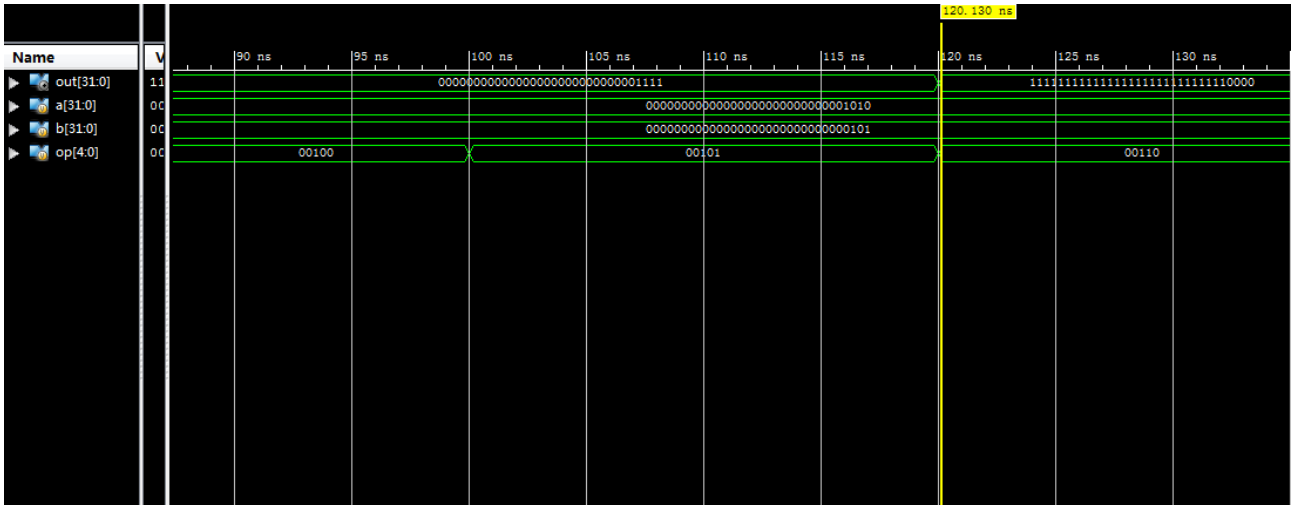
op=3 时， out=32’b00000000000000000000000000000000



op=4 时， out=32’b00000000000000000000000000001111

op=5 时， out=32’b00000000000000000000000000001111

op=6 时， out=32’b11111111111111111111111111110000

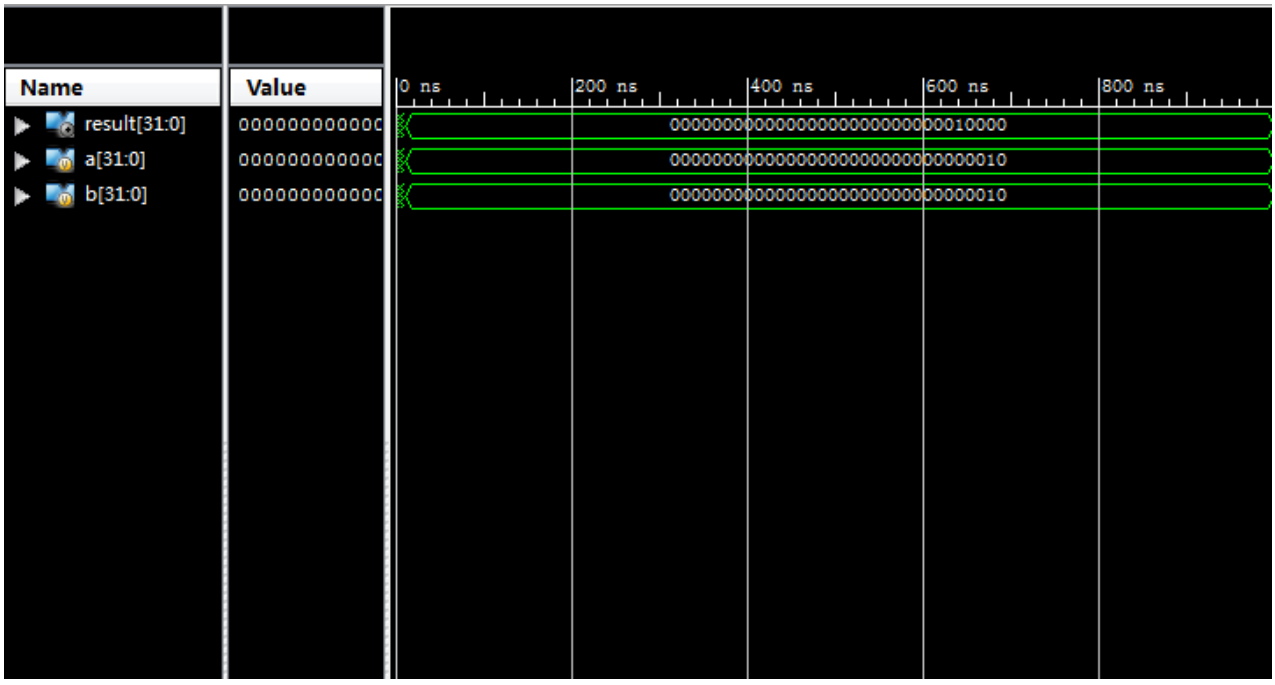


（2）斐波那契数列计算测试

<1>输入 a=2，b=2

得结果为 result=16(=32’b00000000000000000000000000001000)

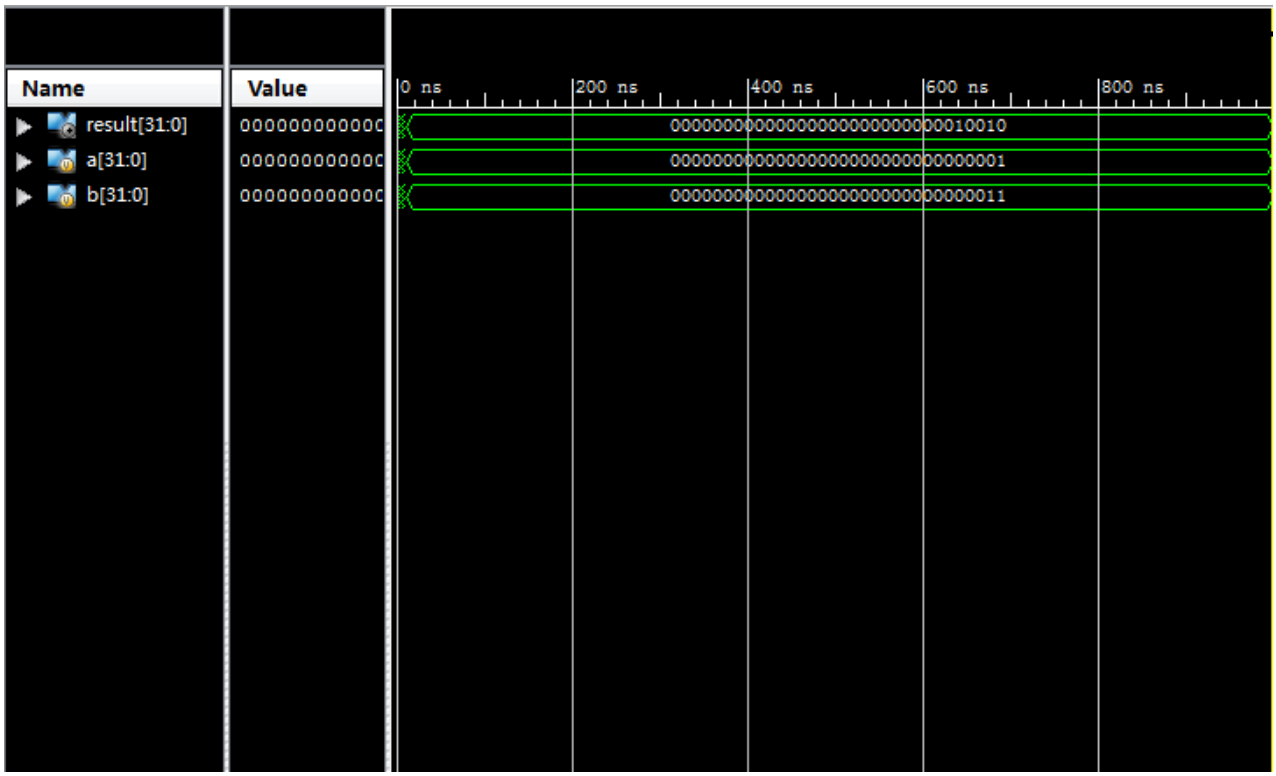
（过程为 2 2 4 6 10 16）



<2>输入 a=1，b=3

得结果为 result=18(=32’b00000000000000000000000000010010)。

（过程为 1 3 4 7 11 18）



附录:

一、 模块源代码

top.v

```
module top(
    input [31:0] a,
    input [31:0] b,
    output [31:0] result
    );
     wire [31:0] tmp[0:4];
     assign tmp[0]=a;
     assign tmp[1]=b;
    alu alu1(tmp[0],tmp[1],5'h01,tmp[2]);
    alu alu2(tmp[1],tmp[2],5'h01,tmp[3]);
    alu alu3(tmp[2],tmp[3],5'h01,tmp[4]);
    alu alu4(tmp[3],tmp[4],5'h01,result);
endmodule
```

alu.v

```
parameter A_NOP =5'h00; //nop
parameter A_ADD =5'h01; //sign_add
parameter A_SUB =5'h02; //sign_sub
parameter A_AND =5'h03; //and
parameter A_OR  =5'h04; //or
parameter A_XOR =5'h05; //xor
parameter A_NOR =5'h06; //nor

module alu(
    input [31:0] alu_a,
    input [31:0] alu_b,
    input [4:0] alu_op,
    output reg [31:0] alu_out
    );
    always@(*)
        case (alu_op)
            A_NOP: alu_out = 0;
            A_ADD: alu_out = alu_a + alu_b;
            A_SUB: alu_out = alu_a - alu_b;
            A_AND: alu_out = alu_a & alu_b;
            A_OR : alu_out = alu_a | alu_b;
            A_XOR: alu_out = alu_a ^ alu_b;
            A_NOR: alu_out = ~(alu_a | alu_b);
            default: alu_out = 0;
        endcase
endmodule
```

toptest.v

```
module toptest;

    // Inputs
    reg [31:0] a;
    reg [31:0] b;

    // Outputs
    wire [31:0] result;

    // Instantiate the Unit Under Test (UUT)
    top uut (
        .a(a),
        .b(b),
        .result(result)
    );

    initial begin


        #10
        // Initialize Inputs
        a = 2;
        b = 2;

        // Wait 100 ns for global reset to finish
        #100;

        // Add stimulus here

    end

endmodule
```

alutest.v

```
module testbunch1();

    // Inputs
    reg [31:0] a,b;
    reg [4:0] op;
    // Outputs
    wire [31:0] out;
    alu alu(a,b,op,out);
```

```
    initial begin
        op=5'h00;

        a=10;
        b=5;
        #20;

        op=5'h01;

        a=10;
        b=5;
        #20;

        op=5'h02;

        a=10;
        b=5;
        #20;

        op=5'h03;

        a=10;
        b=5;
        #20;

        op=5'h04;

        a=10;
        b=5;
        #20;

        op=5'h05;

        a=10;
        b=5;
        #20;

        op=5'h06;

        a=10;
        b=5;
        #20;

    end
endmodule
```