Lab1\_算数运算单元ALU

姓名：姜庆彩

学院：计算机科学与技术学院

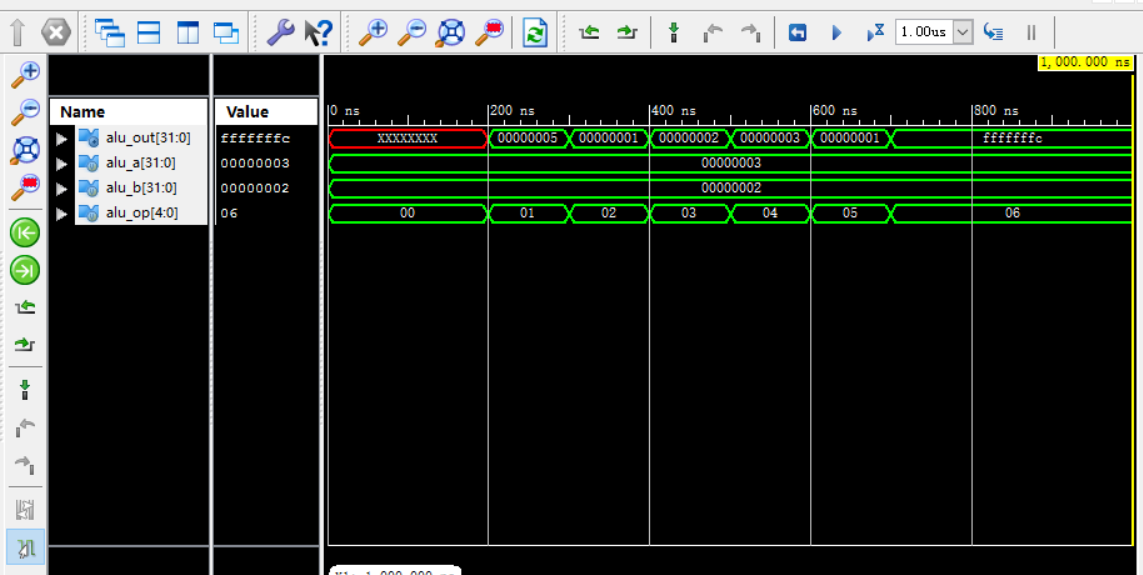
学号：PB15051087

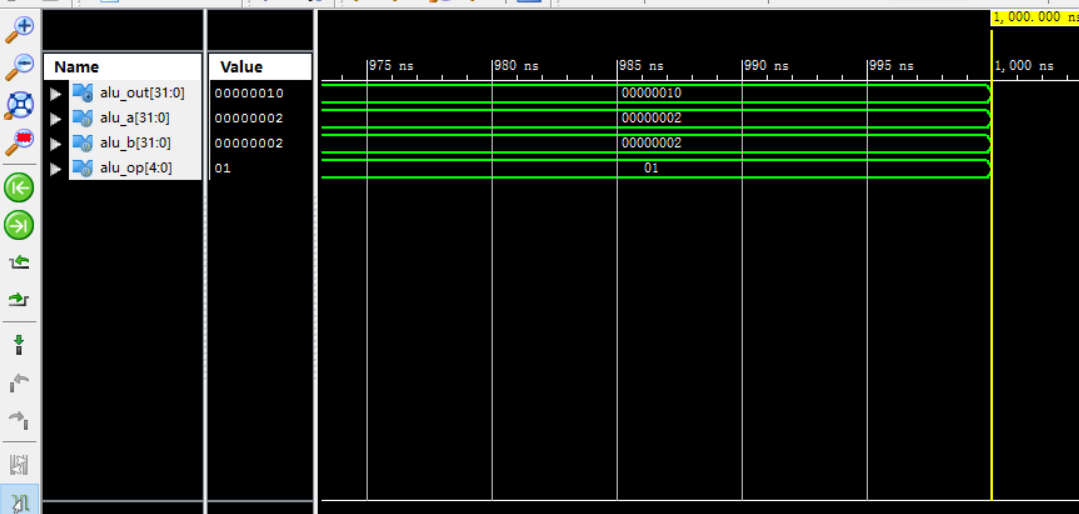
实验要求：

* 设计一算数运算单元ALU
  1. 采用纯组合逻辑设计
  2. 32bit位宽
  3. 完成指定运算功能
* 检查标准：运算结果，仿真结果正确
  1. 1，对所有的7种操作符，设计测试例进行仿真
  2. 2，对于给定的斐波拉契数列运算，进行仿真

我的实现：这次的实验较为简单，就是数字逻辑电路的一个延伸，第一部分七种操作符运算较为简单，而第二部分斐波那契数列的实验我使用了四次例化加运算叠加去完成需要的操作。

实验结果：

仿真图：



源代码：

module alu(

input signed [31:0] alu\_a,

input signed [31:0] alu\_b,

input [4:0] alu\_op,

output reg [31:0] alu\_out

);

always@(\*)

begin

case(alu\_op)

1: alu\_out <= alu\_a + alu\_b;

2: alu\_out <= alu\_a - alu\_b;

3: alu\_out <= alu\_a & alu\_b;

4: alu\_out <= alu\_a | alu\_b;

5: alu\_out <= alu\_a ^ alu\_b;

6: alu\_out <= ~(alu\_a | alu\_b);

default:alu\_out <= alu\_out;

endcase

end

endmodule

module alu\_sim;

reg [31:0] alu\_a;

reg [31:0] alu\_b;

reg [4:0] alu\_op;

wire [31:0] alu\_out;

alu uut (

.alu\_a(alu\_a),

.alu\_b(alu\_b),

.alu\_op(alu\_op),

.alu\_out(alu\_out)

);

initial begin

alu\_a = 3;

alu\_b = 2;

alu\_op = 0;

#100;

alu\_op = 0;

#100;

alu\_op = 1;

#100;

alu\_op = 2;

#100;

alu\_op = 3;

#100;

alu\_op =4;

#100;

alu\_op = 5;

#100;

alu\_op = 6;

end

endmodule

module top (

input signed [31:0] alu\_a,

input signed [31:0] alu\_b,

input [4:0] alu\_op,

output [31:0] alu\_out );

wire [31:0] temp1;

wire [31:0] temp2;

wire [31:0] temp3;

alu a(

.alu\_a(alu\_a),

.alu\_b(alu\_b),

.alu\_op(alu\_op),

.alu\_out(temp1)

);

alu b(

.alu\_a(temp1),

.alu\_b(alu\_b),

.alu\_op(alu\_op),

.alu\_out(temp2)

);

alu c(

.alu\_a(temp2),

.alu\_b(temp1),

.alu\_op(alu\_op),

.alu\_out(temp3)

);

alu d(

.alu\_a(temp3),

.alu\_b(temp2),

.alu\_op(alu\_op),

.alu\_out(alu\_out)

);

Endmodule

module top\_sim;

// Inputs

reg [31:0] alu\_a;

reg [31:0] alu\_b;

reg [4:0] alu\_op;

// Outputs

wire [31:0] alu\_out;

// Instantiate the Unit Under Test (UUT)

top f (

.alu\_a(alu\_a),

.alu\_b(alu\_b),

.alu\_op(alu\_op),

.alu\_out(alu\_out)

);

initial begin

// Initialize Inputs

alu\_a = 2;

alu\_b = 2;

alu\_op = 1;

// Wait 100 ns for global reset to finish

#100;

// Add stimulus here

end

endmodule