

Lab01 ALU 实验报告

PB16051448 赵敏帆

实验目的

- 采用纯组合逻辑设计 32bit ALU
- 用该 ALU 部件完成斐波那契数列计算

实验平台

Vivado 2016.4

实验过程及结果

ALU 部件设计

```
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  
);  
    parameter A_NOP= 5'h00;  
    parameter A_ADD= 5'h01;  
    parameter A_SUB= 5'h02;  
    parameter A_AND= 5'h03;  
    parameter A_OR = 5'h04;  
    parameter A_XOR= 5'h05;  
    parameter A_NOR= 5'h06;  
    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
```

通过预先设定好的 `parameter` 变量，对 `alu_op` 进行译码，并执行相应的操作

top 函数设计

```

module main(
input  [31:0] alu_a,
input  [31:0] alu_b,
output [31:0]alu_out
);
wire [31:0]c;
wire [31:0]d;
wire [31:0]e;
alu alu1(
.alu_a (alu_a),
.alu_b (alu_b),
.alu_op (1),
.alu_out (c)
);
alu alu2(
.alu_a (alu_b),
.alu_b (c),
.alu_op (1),
.alu_out (d)
);
alu alu3(
.alu_a (c),
.alu_b (d),
.alu_op (1),
.alu_out (e)
);
alu alu4(
.alu_a (d),
.alu_b (e),
.alu_op (1),
.alu_out (alu_out)
);
endmodule

```

top 函数通过调用 ALU 来实现斐波那契数列的计算，使用 `c`、`d`、`e` 三个 `wire` 型的变量来传递计算的中间结果，最终使用 `alu_out` 来输出结果。

仿真函数设计

对 ALU 功能仿真

代码

```

module test(

);
reg [31:0]alu_a;
reg [31:0]alu_b;
wire [31:0]alu_out;
reg [4:0]alu_op;
alu test(

```

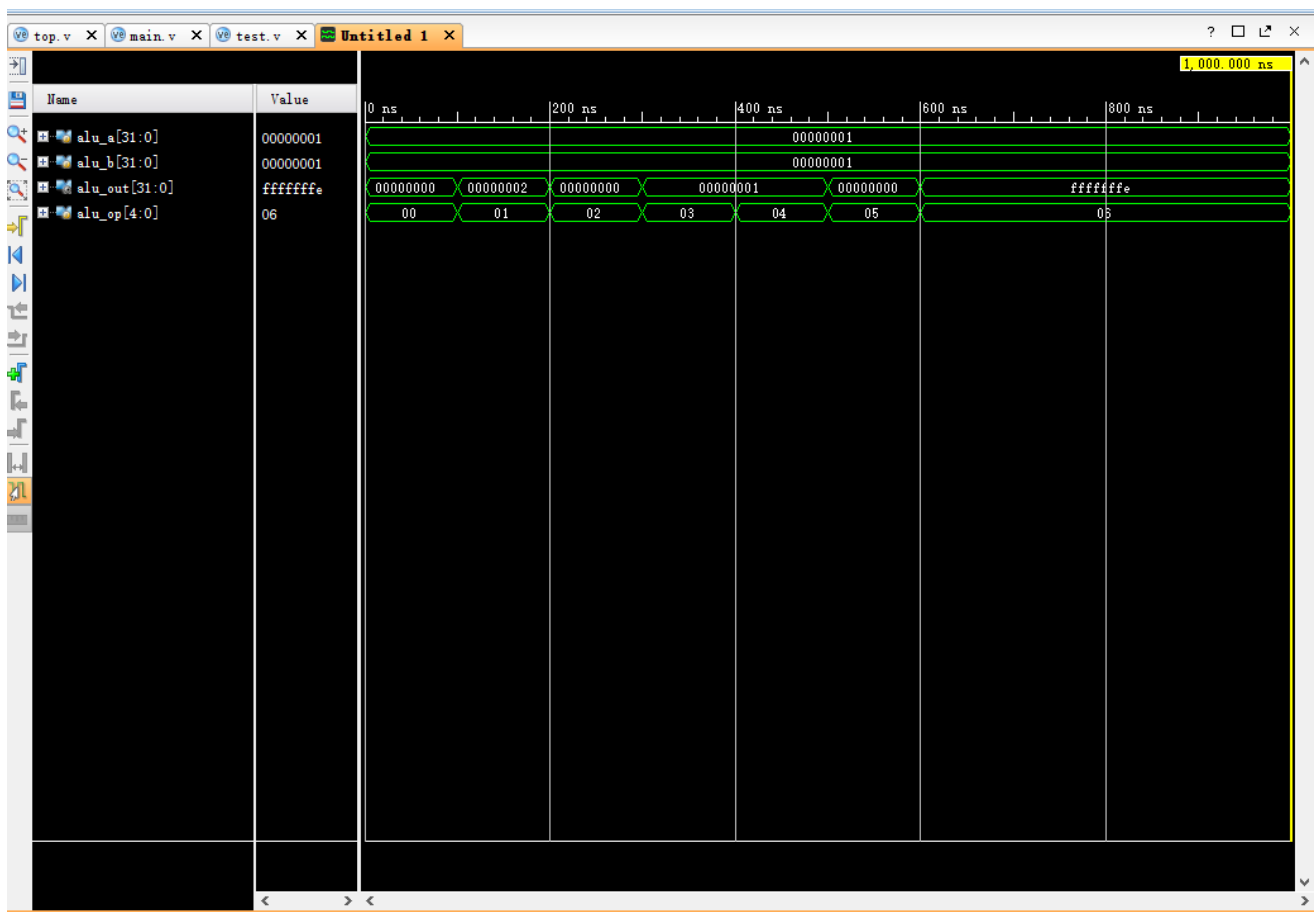
```

.alu_a (alu_a),
.alu_b (alu_b),
.alu_out (alu_out),
.alu_op (alu_op)
);
initial begin
alu_a=32'h1;
alu_b=32'h1;
alu_op=5'h0;
#100
alu_op=5'h1;
#100
    alu_op=5'h2;
#100
        alu_op=5'h3;
#100
            alu_op=5'h4;
#100
                alu_op=5'h5;
#100
                    alu_op=5'h6;

end
endmodule

```

仿真结果



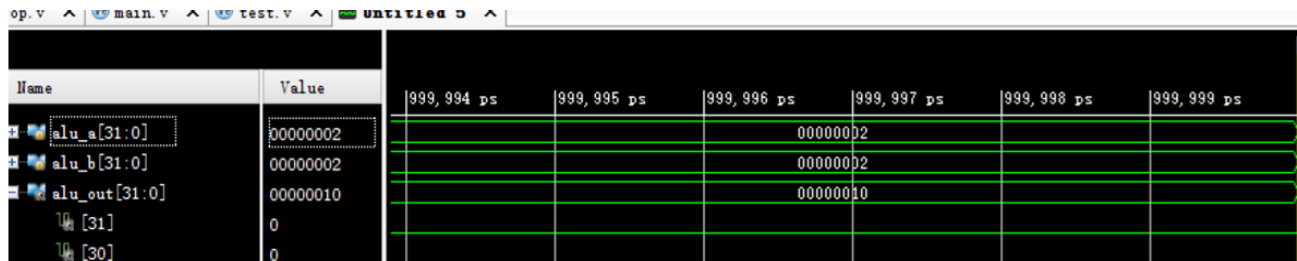
对 top 函数仿真

代码

```
module test(

);
reg [31:0]alu_a;
reg [31:0]alu_b;
wire [31:0]alu_out;
reg [4:0]alu_op;
main test(
.alu_a (alu_a),
.alu_b (alu_b),
.alu_out (alu_out)
);
initial begin
alu_a=32'h2;
alu_b=32'h2;
end
endmodule
```

仿真结果



心得体会

- 通过该实验复习了 verilog 的一些语法知识。
- 仿真时可以选择对 `test` 或是 `top` 进行仿真，选择 `test` 才能得到正确的仿真结果。