

计算机组成原理实验报告

- 实验题目：运算器与寄存器
- 实验日期：2019年3月22日
- 姓名：张劲喲
- 学号：PB16111485
- 成绩：

实验目的：

1. 实现ALU和寄存器部件设计

算术逻辑单元 (ALU)

- **s**：功能选择。加、减、与、或、非、异或等运算
- **a, b**：两操作数。对于减运算，a是被减数；对于非运算，操作数是a
- **y**：运算结果。和、差
- **f**：标志。进位/借位、溢出、零标志

寄存器

- **in, out**：输入、输出数据
- **en, rst, clk**：使能、复位、时钟

2. 实现ALU和寄存器的简单应用

- 求给定两个初始数的斐波拉契数列（结果从同一端口分时输出）

实验设计简述与核心代码：

ALU设计 (ALU.v)

```
module ALU(  
    input signed [5:0] a,    //operand_1 subtracted number when sub  
    input signed [5:0] b,    //operand_2  
    input [2:0] s,          //control_signal  
    output reg [5:0] y,      //result  
    output reg [2:0] f       //signal_flags  
);  
    /*** meaning of flags ***/  
    //f[0]:ZF  
    //f[1]:CF  
    //f[2]:VF  
    //-----  
    /*** operation parameters ***/  
    parameter ADD = 3'b000;
```

```

parameter SUB = 3'b001;
parameter AND = 3'b010;
parameter OR  = 3'b011;
parameter NOT = 3'b100;
parameter XOR = 3'b101;
//-----
parameter LENGTH = 6; //operand bits
reg carry; //save carry signal for VF logic

always @(*) begin
    case(s)
        ADD:
            begin
                {carry,y} = {1'b0,a} + {1'b0,b};
                f[0] = ~(|y);
                f[1] = carry;
                //logic for VF
                f[2] = carry ^ a[LENGTH - 1] ^ b[LENGTH - 1] ^ y[LENGTH - 1];
            end
        SUB:
            begin
                {carry,y} = {1'b0,a} - {1'b0,b};
                f[0] = ~(|y);
                f[1] = carry;
                f[2] = carry ^ a[LENGTH - 1] ^ b[LENGTH - 1] ^ y[LENGTH - 1];
            end
        AND:      begin y = a & b;      f = 0; end
        OR:       begin y = a | b;      f = 0; end
        NOT:      begin y = ~a;        f = 0; end
        XOR:      begin y = a ^ b;      f = 0; end
        default:  begin y = 0;          f = 0; end
    endcase
end
endmodule

```

斐波那契模块设计 (fibonacci.v)

```

module fibonacci(
    input[5:0] a, // init number_1
    input[5:0] b, // init number_2
    input reset,
    input clk,
    output reg [5:0] result
);
//*** inner-transition wires ***
wire[2:0] flag;
wire[5:0] inerResult;
wire[5:0] input1;
wire[5:0] input2;
//-----
reg[5:0] tmp1;
reg[5:0] tmp2;

```

```

assign input1 = tmp1;    // iteration operand_1 (smaller one)
assign input2 = tmp2;    // iteration operand_2 (bigger one)
//-----
/** interface of ALU
module ALU(
input signed [5:0] a,
input signed [5:0] b,
input [2:0] s,
output reg [5:0] y,
output reg [2:0] f
);  **/
ALU alu(input1,input2,3'b000,inerResult,flag);
//-----
always@(posedge reset, posedge clk)
begin
    if(reset)    // reset or initiate
        begin
            tmp1 <= a>b?b:a;
            tmp2 <= a>b?a:b;
            result <= 0;
        end
    if(clk)      // iteration
        begin
            result = inerResult;
            tmp1 = tmp2;
            tmp2 = inerResult;
        end
end

endmodule

```

实验结果：

现场烧录检查：已通过

实现资源消耗与性能统计：

Utilization

Post-Synthesis | Post-Implementation

Graph | Table

Resource	Estimation	Available	Utilization %
LUT	43	63400	0.07
FF	30	126800	0.02
IO	20	210	9.52
BUFG	1	32	3.13

Power analysis from Implemented netlist. Activity derived from constraints files, simulation files or vectorless analysis.

Total On-Chip Power: 18.941 W (Junction temp exceeded!)

Design Power Budget: Not Specified

Power Budget Margin: N/A

Junction Temperature: 111.4°C

Thermal Margin: -26.4°C (-5.5 W)

Effective θ JA: 4.6°C/W

Power supplied to off-chip devices: 0 W

Confidence level: Low

[Launch Power Constraint Advisor](#) to find and fix invalid switching activity

On-Chip Power

97%

95%

- Dynamic: 18.388 W (97%)
- Signals: 0.591 W (3%)
- Logic: 0.286 W (2%)
- I/O: 17.511 W (95%)
- Device Static: 0.552 W (3%)

Design Timing Summary

Setup	Hold	Pulse Width
Worst Negative Slack (WNS): inf	Worst Hold Slack (WHS): inf	Worst Pulse Width Slack (WPWS): NA
Total Negative Slack (TNS): 0.000 ns	Total Hold Slack (THS): 0.000 ns	Total Pulse Width Negative Slack (TPWS): NA
Number of Failing Endpoints: 0	Number of Failing Endpoints: 0	Number of Failing Endpoints: NA
Total Number of Endpoints: 78	Total Number of Endpoints: 78	Total Number of Endpoints: NA

There are no user specified timing constraints.

仿真测试结果:

ALU仿真(test_ALU.v)

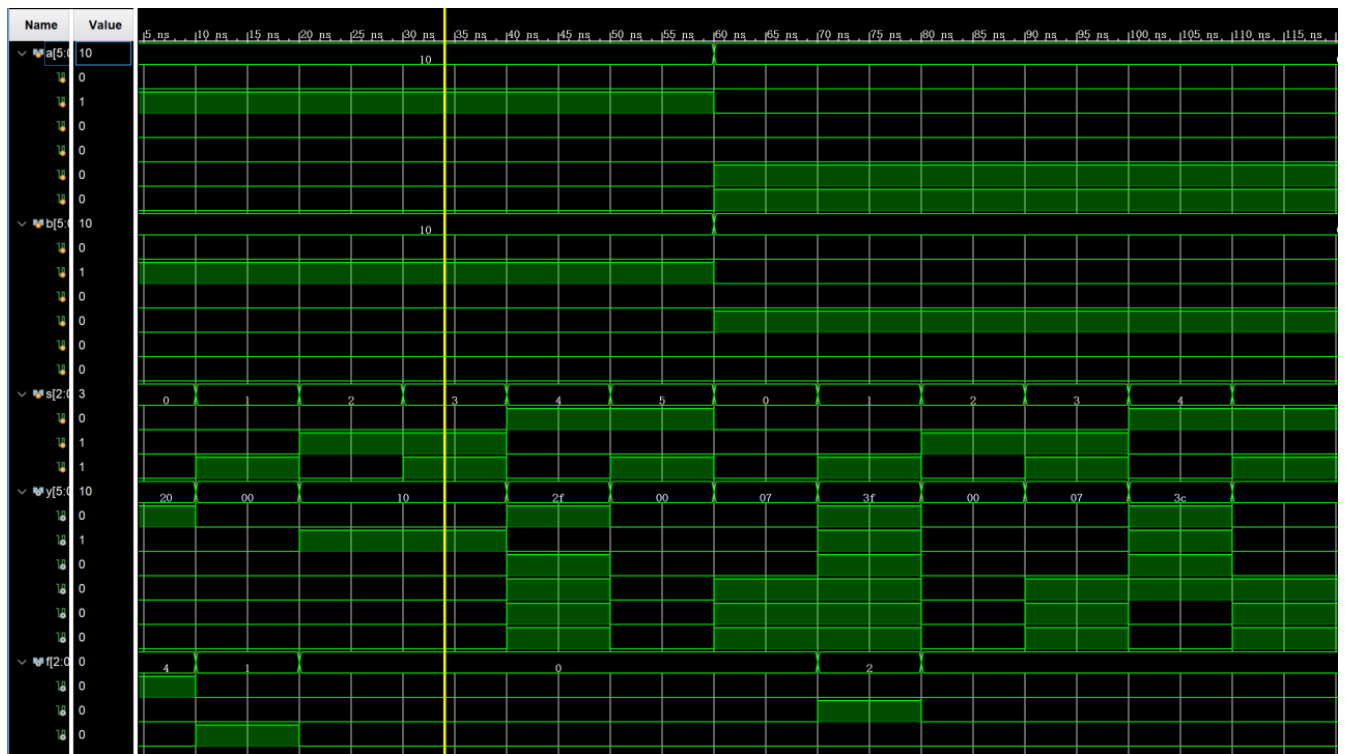
```
module test_ALU(  
  
    );  
    reg [5:0] a,b;  
    reg [2:0] s;  
    wire [5:0] y;  
    wire [2:0] f;  
    ALU alu( .a(a), .b(b), .s(s), .y(y), .f(f));  
    initial  
        begin
```

```

a = 6'b010_000;
b = 6'b010_000;
//signed overflow
s = 3'b000;
#10;
//zero
s = 3'b001;
#10;
s = 3'b010;
#10;
s = 3'b011;
#10;
s = 3'b100;
#10;
s = 3'b101;
#10;
a = 6'b000_011;
b = 6'b000_100;
//carry flag
s = 3'b000;
#10;
//-1(111_111)
s = 3'b001;
#10;
s = 3'b010;
#10;
s = 3'b011;
#10;
s = 3'b100;
#10;
s = 3'b101;
#10;
end
endmodule

```

结果正确：



fibonacci 仿真(test_fibonacci.v)

```

module test_fibonacci(
    );
    reg [5:0] a,b,reset,clk;
    wire [5:0]result;
    fibonacci tf(.a(a),.b(b),.reset(reset),.clk(clk),.result(result));
    initial
        begin
            a = 6'b000_001;
            b = 6'b000_000;
            reset = 0;
            clk = 0;
            #10
            reset = 1;
            clk = 1;
            #10
            reset = 0;
            clk = 0;
            #10
            clk = 1;
            #10
            clk = 0;
            #10
            clk = 1;
            #10
            clk = 0;
            #10
            clk = 1;
            #10
            clk = 0;
        end

```

```

#10
clk = 1;
#10
clk = 0;
#10
clk = 1;
#10
clk = 0;
#10
clk = 1;
#10
clk = 0;
#10
clk = 1;
#10
clk = 0;
#10
clk = 1;
end
endmodule

```

结果正确：



实验总结与感想：

1. 通过实验了解了ALU的设计实现，了解了ALU和符号位的简单应用。
2. 复习了Verilog语法，提高了编程实践能力。