

实验报告

实验题目：lab1 运算器与寄存器 日期：2019/3/22

姓名：__田宏宇__ 学号：_PB17111573_ 成绩：_____

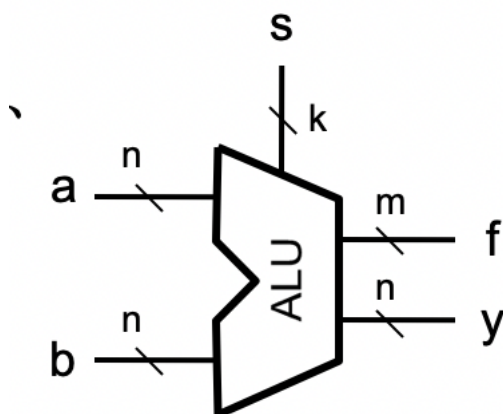
实验目的：

- 熟练Vivado设计实现流程
- 模块化、层次化、参数化设计方法
- 组合逻辑电路和寄存器的描述方法

逻辑设计：

1. ALU：根据功能选择s，对a和b进行算术（加、减）或者逻辑（与、或、非、异或）运算，产生运算结果y和相应标志f（进位/借位、溢出、零标志）。对于算术运算，影响进位/借位、溢出、零标志；对于逻辑运算，仅零标志有效。

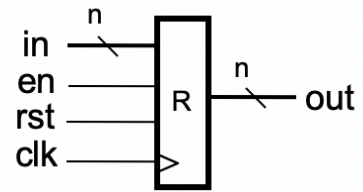
f: 标志位，包括进位/借位(CF)，符号位(S)，溢出位(V)，零标志(Z)



2.

寄存器

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



3.

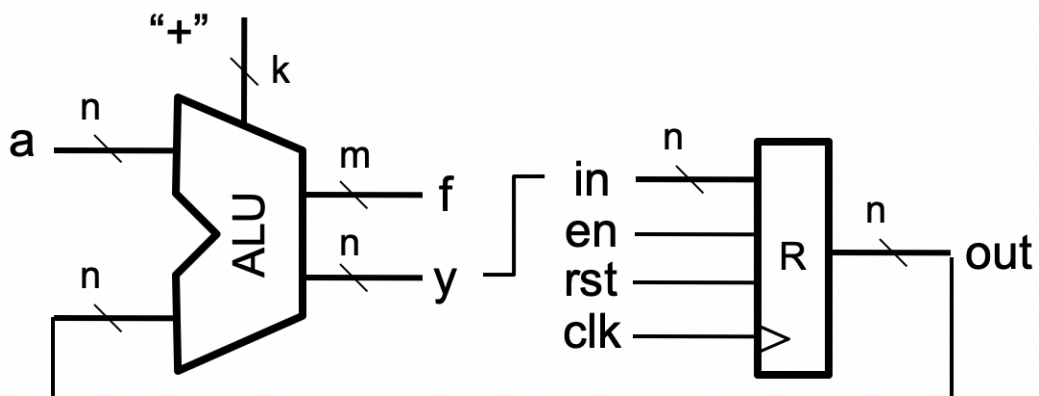
a. 比较两个数大小关系：已在 1 中实现，为 ALU 的 07 功能

when $a > b$: $y=1$ $f=0$;

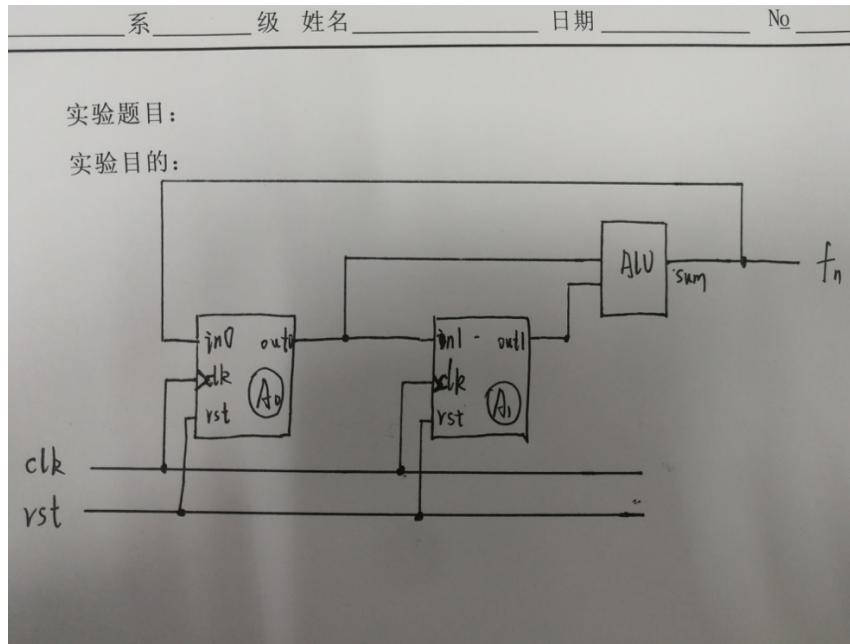
when $a < b$: $y=0$ $f=0$;

when $a = b$: $y=0$ $f=1$.

b. 求多个数累加和



c. 求给定两个初始数的斐波拉契数列



具体实现

1. ALU 实现

```

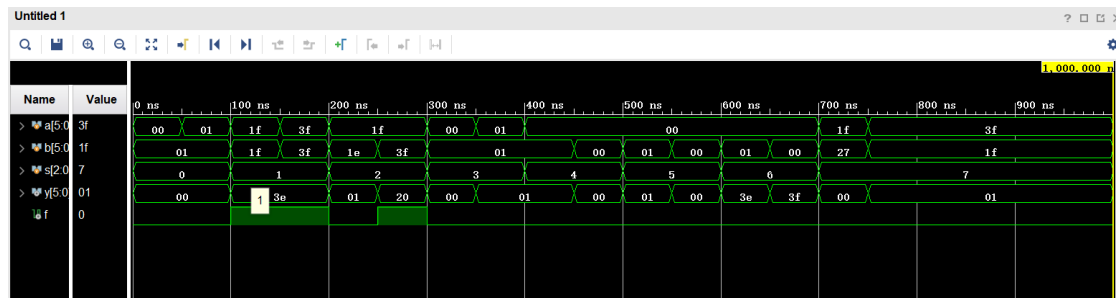
22
23 module ALU(
24     input      [5:0]  a,
25     input      [5:0]  b,
26     input      [2:0]  s,
27     output     reg [5:0] y,
28     output     reg [2:0] f
29 );
30
31
32 parameter A_NOP = 3'h00;
33 parameter A_ADD = 3'h01;
34 parameter A_SUB = 3'h02;
35 parameter A_AND = 3'h03;
36 parameter A_OR  = 3'h04;
37 parameter A_XOR = 3'h05;
38 parameter A_NOR = 3'h06;
39 parameter A_LAT = 3'h07;
40
41 always@(*)
42 begin
43     case (s)
44     A_NOP:
45         begin y = 6'b0; f = 0; end
46     A_ADD:
47         begin
48             y = a + b;
49             if ((a[5]==0&&b[5]==0&a[4]==1&b[4]==1)//两正数相加溢出
50                 |(a[5]==1&&b[5]==1&a[4]==1&b[4]==1))// 两负数相加溢出
51                 f=1;
52             else f=0;
53             end
54     A_SUB:
55         begin
56             y = a - b;
57             if ((a[5]==1&&b[5]==0&a[4]==1&b[4]==1)//不同符号数相减溢出
58                 |(a[5]==0&&b[5]==1&a[4]==1&b[4]==1))
59                 f=1;
60             else f=0;
61             end
62     A_AND: begin y = a & b; f=0;end
63     A_OR : begin y = a | b; f=0;end
64     A_XOR: begin y = a ^ b; f=0;end
65     A_NOR: begin y = a ^^ b; f=0;end
66     A_LAT:
67         begin
68             if (a > b)
69                 begin y=1; f=0; end
70             else if (a < b)
71                 begin y=0; f=0; end
72             else
73                 begin y=0; f=1; end//when equal f=1
74             end
75     endcase
76 end
77 endmodule

```

仿真代码如下：

```
23 module ALU_test();
24     reg [5:0] a,b;
25     reg [2:0] s;
26     wire [5:0] y;
27     wire f;
28     ALU LUT (
29         .a(a),
30         .b(b),
31         .s(s),
32         .y(y),
33         .f(f)
34     );
35     initial
36     begin
37         /*0空运算*/
38         s=3'b0000;a=6'b000000; b=6'b000001;#50;
39         s=3'b0000;a=6'b000001; b=6'b000001;#50;
40         /*1算术加运算*/
41         s=3'b0001;a=6'b011111; b=6'b011111;#50;
42         s=3'b0001;a=6'b111111; b=6'b111111;#50;
43         /*2算术减运算*/
44         s=3'b0010;a=6'b011111; b=6'b011110;#50;
45         s=3'b0010;a=6'b011111; b=6'b111111;#50;
46         /*3按位与*/
47         s=3'b0011;a=6'b000000; b=6'b000001;#50;
48         s=3'b0011;a=6'b000001; b=6'b000001;#50;
49         /*4按位或*/
50         s=3'b0100;a=6'b000000; b=6'b000001;#50;
51         s=3'b0100;a=6'b000000; b=6'b000000;#50;
52         /*5按位异或*/
53         s=3'b0101;a=6'b000000; b=6'b000001;#50;
54         s=3'b0101;a=6'b000000; b=6'b000000;#50;
55         /*6按位或非*/
56         s=3'b0110;a=6'b000000; b=6'b000001;#50;
57         s=3'b0110;a=6'b000000; b=6'b000000;#50;
58         /*7a>b时,输出1,否则输出0*/
59         s=3'b0111;a=6'b011111; b=6'b100111;#50;
60         s=3'b0111;a=6'b111111; b=6'b011111;#50;
61     end
62
63 endmodule
64
```

仿真结果如下



2. 寄存器

```

23 module REG_file(
24     input    wire  clk,
25     input    wire  rst,
26     input    wire  en,
27     input    wire [5:0] in,
28     output   reg  [5:0] out
29 );
30 always @ (posedge clk or posedge rst)
31 if(rst ==1)
32     out<=0;
33 else if(en==1)
34     out<=in;
35 endmodule
36

```

3.

a. 在 ALU 中实现（见 ALU 仿真图 700ns—900ns）

c. FIB

```

22 |
23 | module FIB(
24 |     input en,
25 |     input [5:0]f0,
26 |     input [5:0]f1,
27 |     input rst,
28 |     input clk,
29 |     output reg [5:0]fn
30 | );
31 |     wire [2:0]f;
32 |     wire [5:0] in0;
33 |     wire [5:0] out0;
34 |     wire [5:0] in1;
35 |     wire [5:0] out1;
36 |     wire [5:0] sum;
37 |
38 |     //assign fn=sum;
39 |     ALU ADD (out0,out1,3'h01,sum ,f);
40 |     REG_file A0 (clk,rst,1,in0,out0);
41 |     REG_file A1 (clk,rst,1,in1,out1);
42 |     assign in0=(en)?f0:sum;
43 |     assign in1=(en)?f1:out0;
44 |     always@(posedge clk)
45 |     begin
46 |         fn<=sum;
47 |     end
48 | endmodule
49 |

```

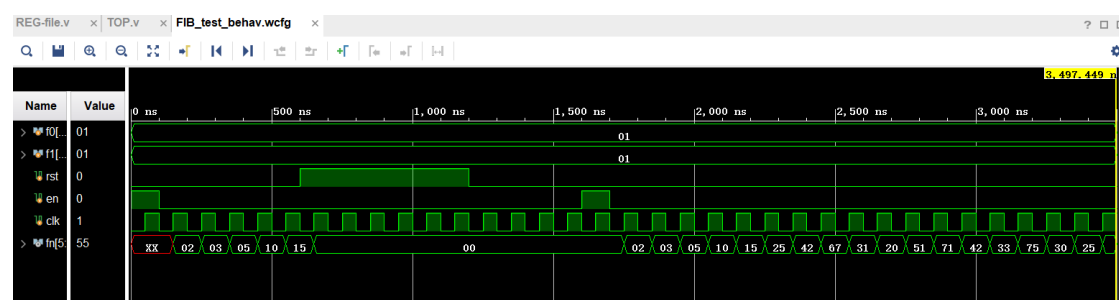
仿真代码如下

```

22
23 module FIB_test();
24     reg [5:0] f0;
25     reg [5:0] f1;
26     reg rst;
27     reg en;
28     reg clk;
29     wire [5:0]fn;
30     FIB ALU1 (
31         .en(en),
32         .f0(f0),
33         .f1(f1),
34         .rst(rst),
35         .clk(clk),
36         .fn(fn)
37     );
38     initial f0=6'b000001;
39     initial f1=6'b000001;
40     initial clk=0;
41     initial rst=0;
42     initial
43     begin
44         en=1;
45         #100 en=0;
46         #1500 en=1;
47         #100 en=0;
48     end
49     always #50 clk=~clk;
50     initial begin
51         #600 rst=1;
52         #600 rst=0;
53     end
54
55 endmodule

```

仿真结果如下



实验总结:

在本次实验中，我学会了如何设计算术逻辑单元（ALU）、寄存器、

以及利用 ALU 和寄存器设计逻辑电路求给定两个初始数的贝波那契数列，组成原理实验是模拟数字电路实验的延伸，通过这次实验我复习了 verilog 的语法与编程技巧，熟悉了 vivado2018 的操作环境，这对我完成后续的组成原理实验有很大的帮助。