Lab1 运算器及其应用

梁峻滔　PB19051175

2021 年 4 月 13 日

一、算术逻辑单元ALU

1. 逻辑设计

   ALU的模块和端口如图1所示，各端口的作用如下：

   - f：指定操作类型，加、减、与、或、异或等运算
   - a，b：操作数，对于减运算，a是被减数
   - y：运算结果
   - z：零标志，若运算结果为零，则z置1

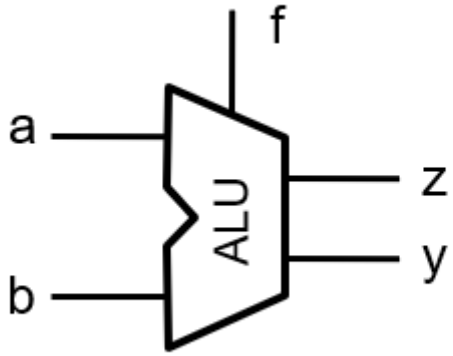


图1：ALU模块

ALU 模块功能表

f	y	z
000	a + b	*
001	a - b	*
010	a & b	*
011	a | b	*
100	a ^ b	*
其他	0	1

图2

2. 核心代码

   - ALU模块
     - 设计文件

```
module ALU_module
    #(parameter WIDTH=6)
     (output reg[WIDTH-1:0] y,
      output reg z,
      input[WIDTH-1:0] a,b,
      input[2:0] f);
assign z = (y == 0);
always @(*)
begin
    case(f)
        3'b000: y = a + b;
        3'b001: y = a - b;
        3'b010: y = a & b;
        3'b011: y = a | b;
        3'b100: y = a ^ b;
        default: y = 0;
    endcase
end
endmodule
```

- 仿真文件

```
module ALU_simulation();
reg [31:0] a,b;
reg [2:0] f;
wire [31:0] y;
wire z;
ALU_module #(32) alu1(y,z,a,b,f);   //32位ALU
initial
begin
        a = 32'h1; b = 32'h1; f = 3'h0;
    #20 a = 32'h2; b = 32'h2; f = 3'h1;
    #20 a = 32'h0; b = 32'h1; f = 3'h2;
    #20 a = 32'h1; b = 32'h0; f = 3'h3;
    #20 a = 32'h1; b = 32'h1; f = 3'h4;
    #20 $finish;
end
endmodule
```

- 6位ALU下载测试
  - 设计文件

```
module decoder
    (input en,
     input[1:0] sel,
     output ef,ea,eb);
assign ef = en & ~sel[1] & ~sel[0];
assign ea = en & ~sel[1] & sel[0];
assign eb = en & sel[1] & ~sel[0];
endmodule

module distributor
    #(parameter D_WIDTH = 6)
    (input[D_WIDTH-1:0] d,
     input[1:0] sel,
     output reg[2:0] f,
```

```
     output reg[D_WIDTH-1:0] a,b);
always @(*)
begin
    case(sel)
        2'b00: f = d[2:0];
        2'b01: a = d;
        2'b10: b = d;
        default: begin
          f  = 3'b111;
          a = 0;
          b = 0;
        end
    endcase
end
endmodule

module register
    #(parameter R_WIDTH = 6)
    (input[R_WIDTH-1:0] d,
     input en,
     input clk,
     output reg[R_WIDTH-1:0] op);
always @(posedge clk)
    if(en) op <= d;
endmodule

module register1
    (input d,
     input en,
     input clk,
     output reg op);
always @(posedge clk)
    if(en) op <= d;
endmodule

module ALU
    #(parameter DATA_WIDTH = 32)
(
     input en,clk,
     input[1:0] sel,
     input[DATA_WIDTH-1:0] d,
     output[DATA_WIDTH-1:0] y,
     output z
);
wire ef,ea,eb;
wire[2:0] f;
wire[DATA_WIDTH-1:0] a,b;
wire iszero;
wire[2:0] opf;
wire[DATA_WIDTH-1:0] result,opa,opb;
decoder decoder1(en,sel,ef,ea,eb);
distributor #(DATA_WIDTH) dis(d,sel,f,a,b);
register #(DATA_WIDTH) A(a,ea,clk,opa);
register #(DATA_WIDTH) B(b,eb,clk,opb);
register #(3) F(f,ef,clk,opf);
ALU_module #(DATA_WIDTH) alu(result,iszero,opa,opb,opf);
register #(DATA_WIDTH) Y(result,ef,clk,y);
register1 Z(iszero,ef,clk,z);
```

```
endmodule
```

3. 仿真结果

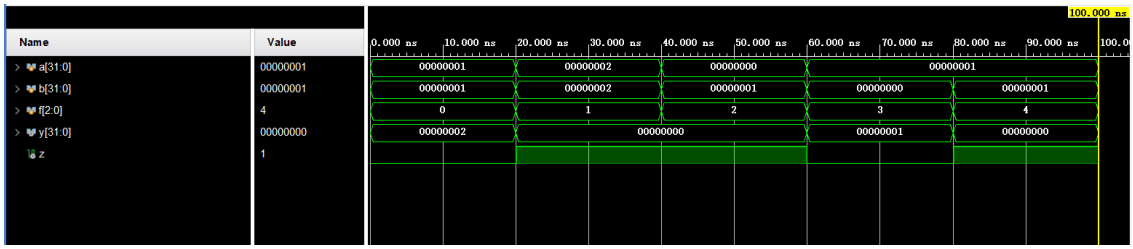


图3：32位ALU仿真波形

如图3所示，当a = 1，b = 1，f = 0(加操作)时，输出结果y = 2，z = 0，运算结果正确；同样，当a，b变化，后续 f 对应操作的运算结果也都正确。

4. 电路图、电路资源和时间性能报告

- 数据通路

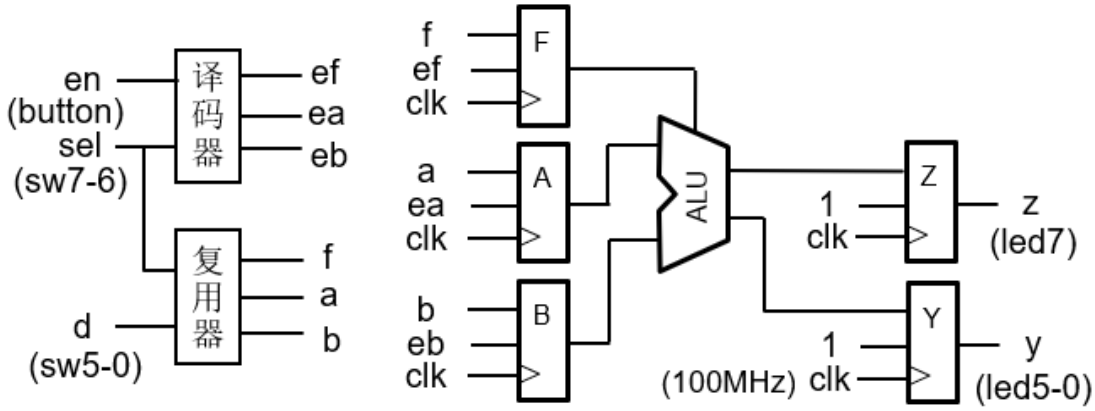


图4：ALU下载测试数据通路

该数据通路由ALU模块、2-4译码器、复用器和寄存器组成，用于在 fpgaol 上测试ALU的功能。en是输入使能， sel是数据选择信号，d为数据输入。

- RTL电路图

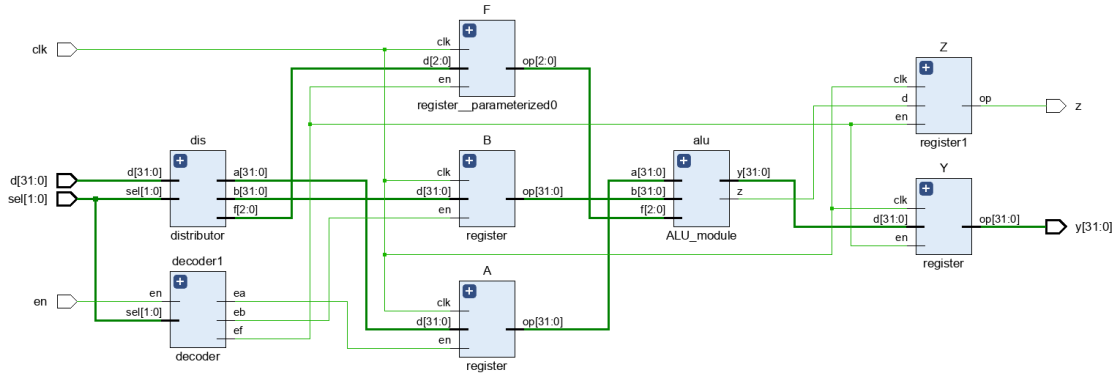


图5：32位ALU的RTL电路图

- 综合电路图

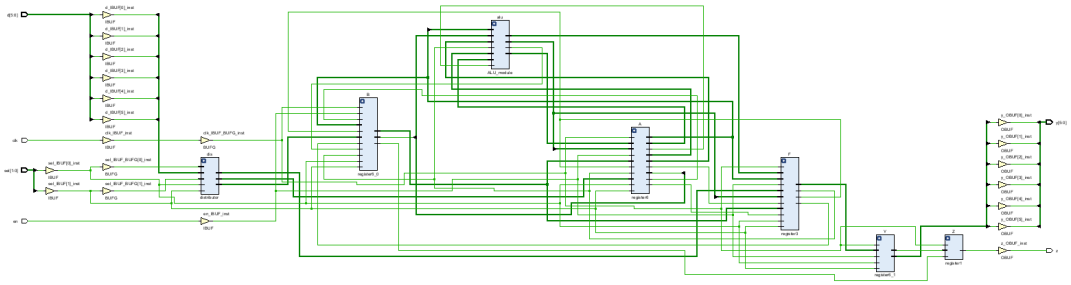


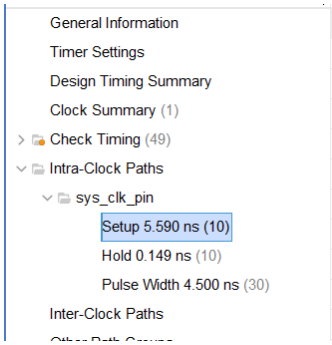
图6：综合电路图

- 综合电路资源报告

Name	Slice LUTs (63400)	Slice Registers (126800)	Bonded IOB (210)	BUFGCTRL (32)
N ALU	54	37	17	3
A (register6)	16	6	0	0
alu (ALU_module)	3	0	0	0
B (register6_0)	13	6	0	0
dis (distributor)	8	15	0	0
F (register3)	13	3	0	0
Y (register6_1)	1	6	0	0
Z (register1)	0	1	0	0

图7：综合电路资源

- 时间性能报告



Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay
Path 1	5.590	5	6	9	A/op_reg[1]/C	Z/op_reg/D	4.274
Path 2	6.209	5	6	9	A/op_reg[1]/C	Y/op_reg[5]/D	3.655
Path 3	6.466	5	6	9	A/op_reg[1]/C	Y/op_reg[4]/D	3.398
Path 4	6.598	4	5	9	A/op_reg[1]/C	Y/op_reg[3]/D	3.266
Path 5	6.652	4	5	9	A/op_reg[0]/C	Y/op_reg[1]/D	3.212
Path 6	6.665	4	5	9	A/op_reg[1]/C	Y/op_reg[2]/D	3.199
Path 7	6.979	4	5	9	A/op_reg[0]/C	Y/op_reg[0]/D	2.885
Path 8	8.723	1	2	18	F/op_reg[0]/C	F/op_reg[0]/D	1.141
Path 9	8.727	1	2	15	F/op_reg[1]/C	F/op_reg[1]/D	1.137
Path 10	8.739	1	2	9	F/op_reg[2]/C	F/op_reg[2]/D	1.125

图8：时间性能报告

5. 结果分析和总结

- 6位ALU下载测试结果
  - 加操作：令a = 1，b = 5，执行加运算后的结果为6，如图9所示：

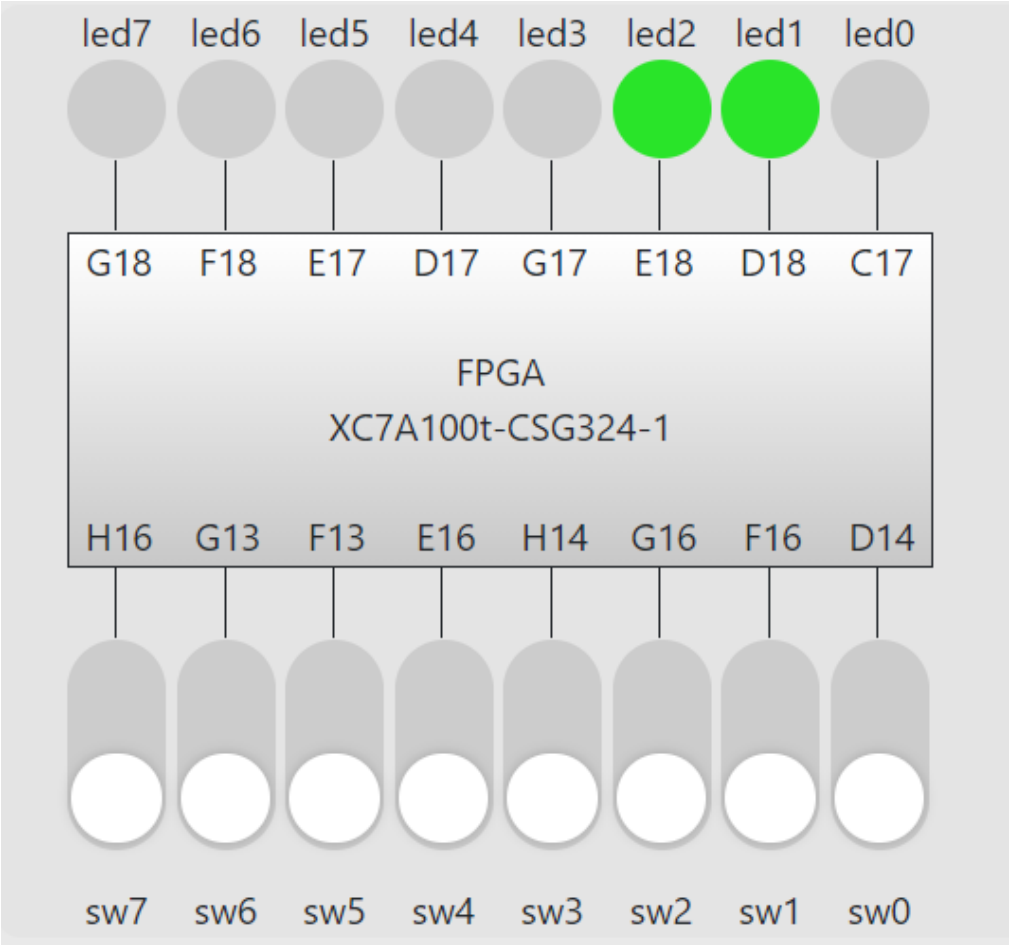


图9：1 + 5 = 6

- 减操作：令a = 1，b = 1，执行减运算后的结果为0，led7(z)亮起：

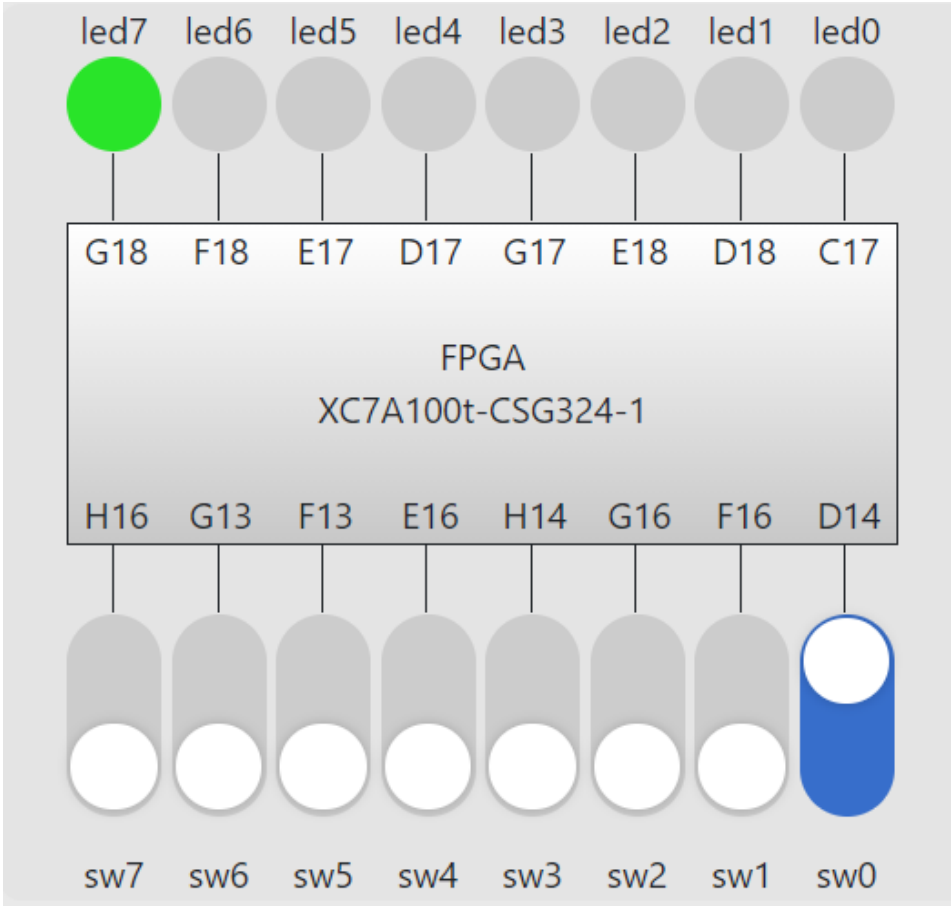


图10：1 - 1 = 0

- 异或操作：令a = 3，b = 3，执行异或运算后结果为0：

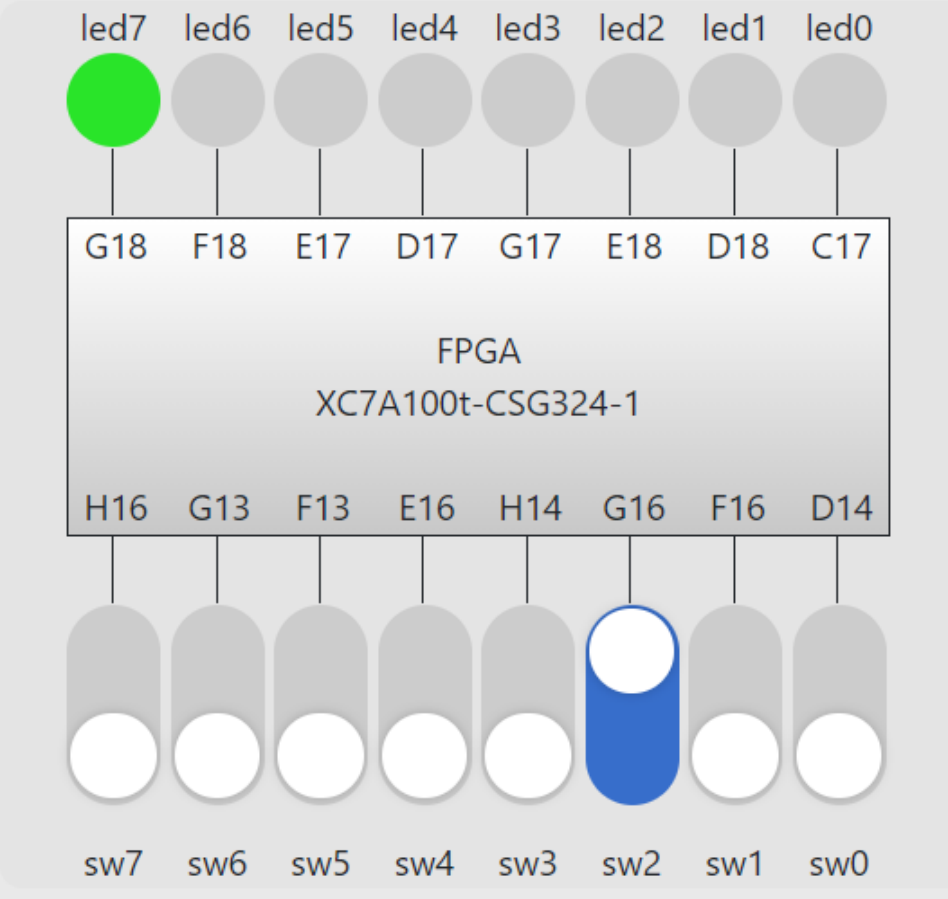


图11：3 ^ 3 = 0

其他操作和改变操作数也能得出正确结果。

二、ALU应用：计算斐波那契-卢卡斯数列

1. 逻辑设计
   - 状态图：

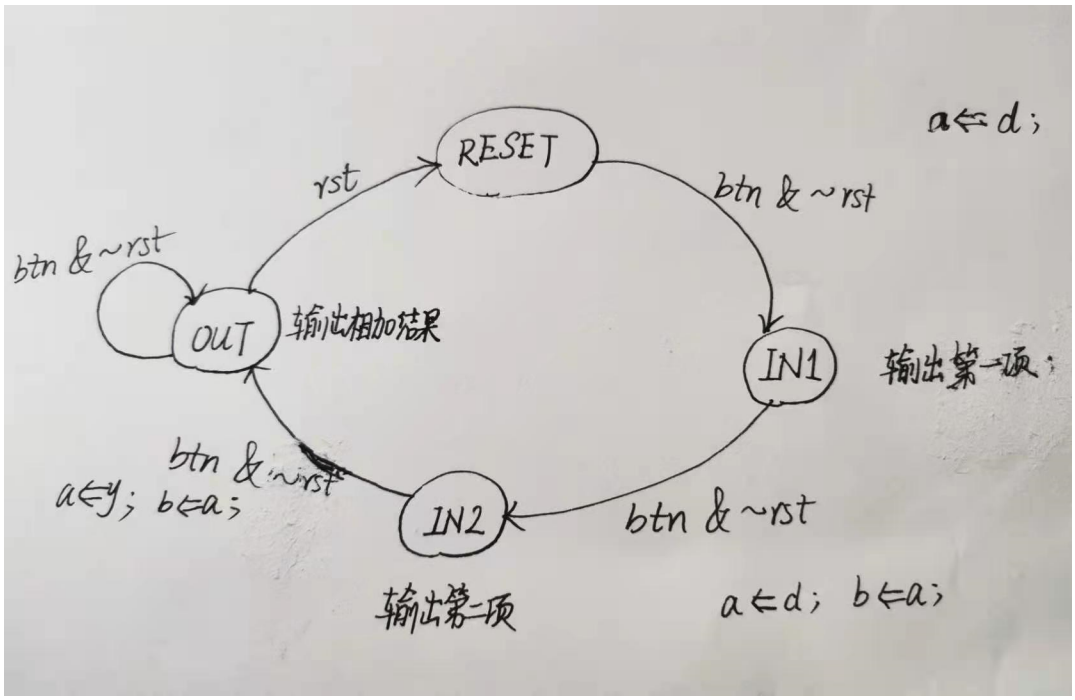


图12：FLS状态图

- RESET：初始状态/复位状态
- IN1：已输入数列第一项

- IN2：已输入数列第二项
- OUT：持续输出数列的后续项

- 数据通路

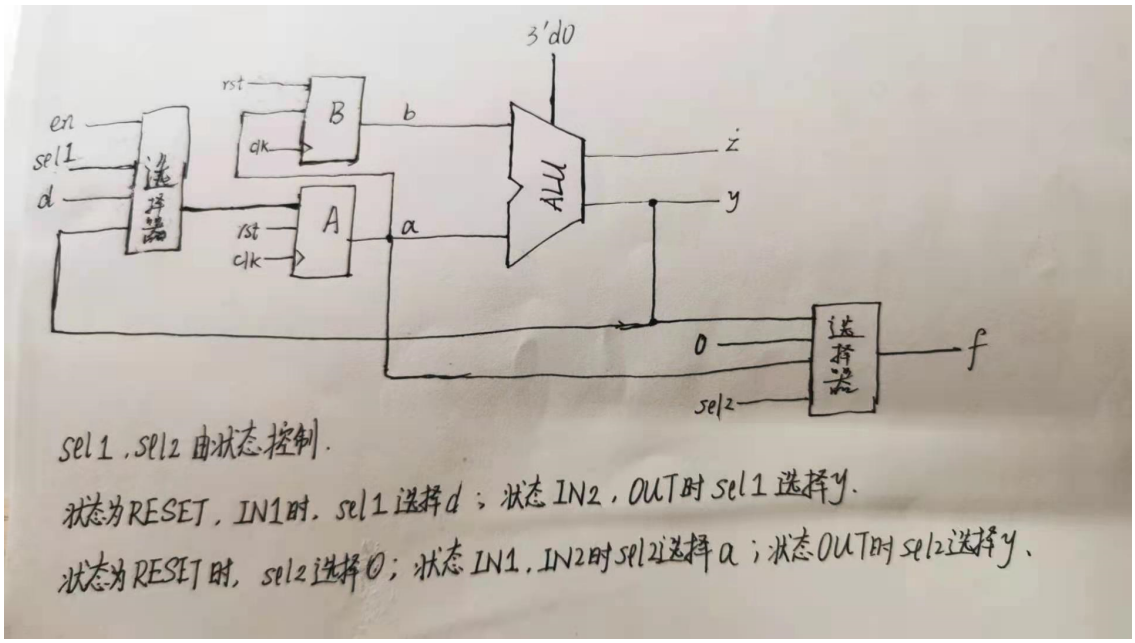


图13：FLS数据通路

2. 核心代码

- 设计文件

```
module FLS(
    input clk,rst,en,
    input [6:0] d,
    output reg [6:0] f
);
//取信号边沿
reg button_r1,button_r2;
wire button_edge;
always @(posedge clk)
    button_r1 <= en;
always @(posedge clk)
    button_r2 <= button_r1;
assign button_edge = button_r1 & (~button_r2);
//参数和变量设置
parameter RESET = 2'b00;
parameter IN1 = 2'b01;
parameter IN2 = 2'b10;
parameter OUT = 2'b11;
parameter ADD = 3'b000;
wire z;
wire [6:0]y;
reg [1:0] cs,ns;
reg [6:0]a,b;
ALU_module #(7) alu(a,b,ADD,y,z);
//有限状态机第一部分
always @(*)
begin
    if(rst == 0)
    begin
        case(cs)
            RESET: ns = IN1;
```

```
            IN1: ns = IN2;
            IN2: ns = OUT;
            OUT: ns = OUT;
            default: ns = RESET;
        endcase
    end
    else ns = RESET;
end
//有限状态机第二部分
always @(posedge clk)
begin
    if(rst) begin cs <= RESET; a <= 7'b0; b <= 7'b0; end
    else if(button_edge)
        begin
            cs <= ns;
            case(cs)
                RESET: a <= d;
                IN1: begin a <= d; b <= a; end
                OUT: begin a <= y; b <= a; end
            endcase
        end
end
//有限状态机第三部分
always @(*)
begin
    case(cs)
        RESET: f = 7'b0;
        IN1: f = a;
        IN2: f = a;
        OUT: f = y;
        default: f = 7'b0;
    endcase
end
endmodule
```

- 仿真文件

```
module ALU_simulation();
reg [31:0] a,b;
reg [2:0] f;
wire [31:0] y;
wire z;
ALU_module #(32) alu1(y,z,a,b,f);
initial
begin
        a = 32'h1; b = 32'h1; f = 3'h0;
    #20 a = 32'h2; b = 32'h2; f = 3'h1;
    #20 a = 32'h0; b = 32'h1; f = 3'h2;
    #20 a = 32'h1; b = 32'h0; f = 3'h3;
    #20 a = 32'h1; b = 32'h1; f = 3'h4;
    #20 $finish;
end
endmodule
```

3. 仿真结果

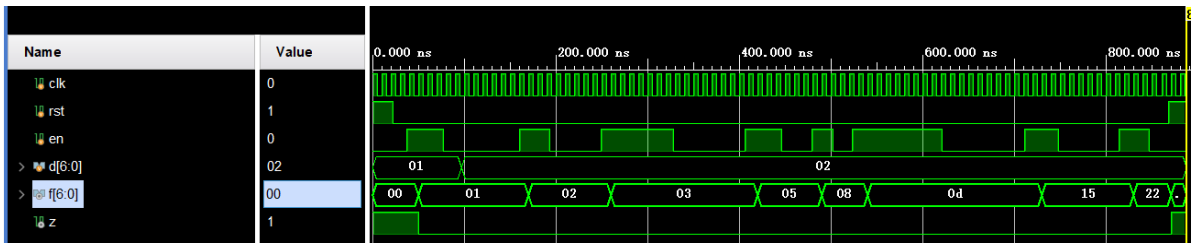


图14：FLS仿真波形

如图14所示，在每次按下button(en高电平)时，输出(f)能正确显示斐波那契数列的各项。

4. 下载测试结果

下载到 fpgaol 上运行的结果与仿真一致，在每次按下button后，led能依次显示各项。

三、意见和建议

计组一门课包含理论课和实验课内容，理论课要复习和写作业，实验内容本身难度不小，需要较长时间才能完成，实验报告的内容要求得也很多，而且在撰写报告过程中还有排版和格式问题，这样一来，学生个人花费在计组一门课的时间就会巨多。希望能在实验报告方面降低要求，比如减少报告要求撰写的条项(如查看电路图、电路资源等贴图需要花费很多时间)、数据通路适当给提示等，就数据通路而言，设计出状态后即可编写设计文件代码，但要画出结构化的数据通路又要在此基础上考虑电路的更多模块和连接问题，难度大而且耗费很多时间。

四、设计和测试文件原代码

见附件。