中国科学技术大学计算机学院

《计算机组成原理实验报告》

实验题目：运算器及其应用

学生姓名：林宸昊

学生学号：PB20000034

完成日期：2022.3.15

【实验题目】运算器及其应用

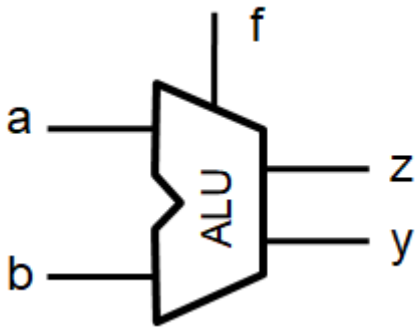
【实验目的】 设计并实现ALU，并在后续中能够利用前述ALU进行复杂设计

【实验平台】 VIVADO FPGAOL

【实验内容】

【一：ALU模块的逻辑设计与仿真】

- 数据通路：



- 状态图：

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

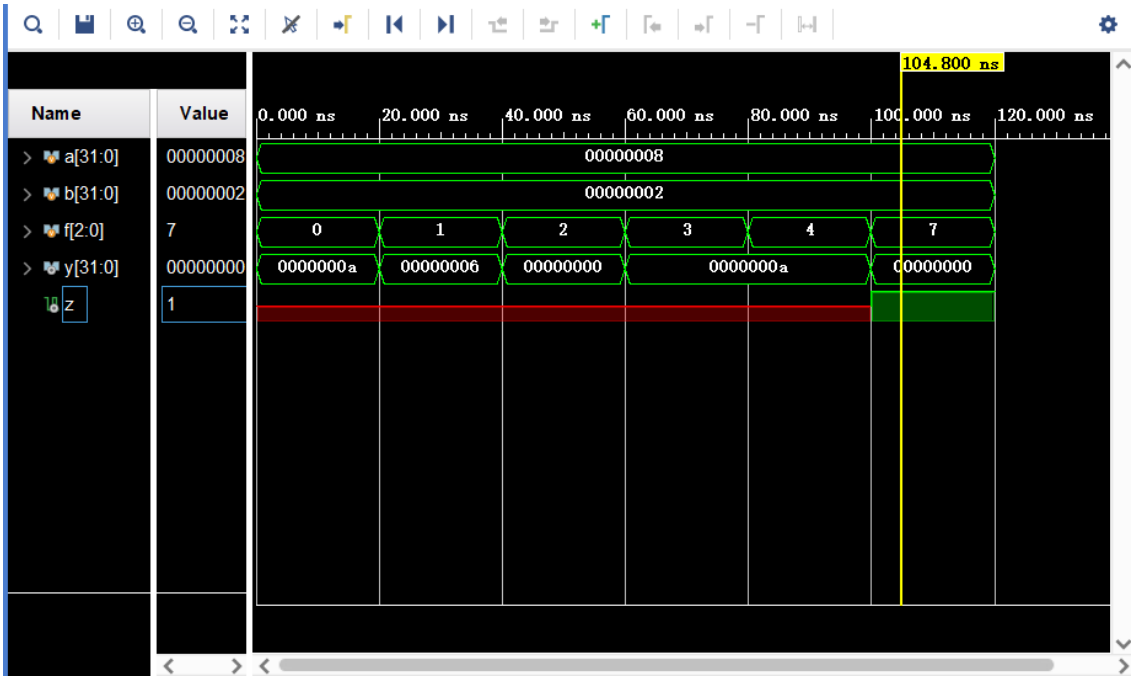
- 编写设计文件：

```
module alu_32(
    input [31:0] a, b,
    input [2:0] f,
    output reg [31:0] y,
    output reg z
    );
    always@(*)
    begin
        case(f)     //通过case语句实现操作功能的选择
        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:
            begin
                y = 0;
                z = 1;
            end
        endcase
    end
endmodule
```

- 仿真文件

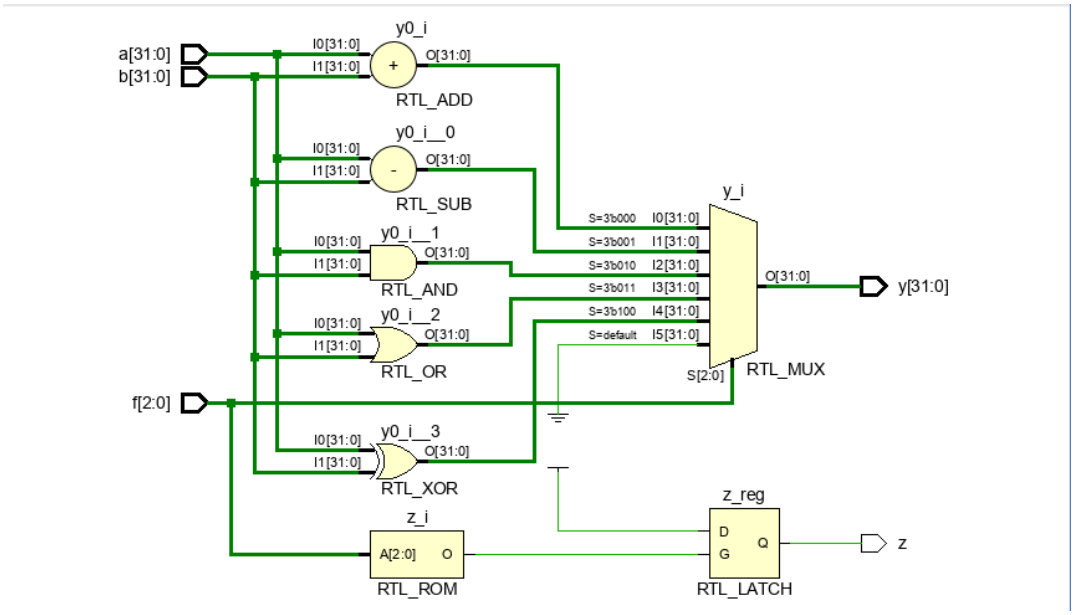
```
module sim1(

    );
    reg [31:0] a,b;
    reg [2:0] f;
    wire [31:0] y;
    wire z;          //输出必须是线网型
    alu_32 alu(a, b, f, y, z);
    initial
    begin
        a = 32'h8; b = 32'h2;
        f = 3'b0; #20 f = 3'b1; #20 f = 3'b010; #20 f = 3'b011;
        #20 f = 3'b100; #20 f = 3'b111; #20 $finish;
    end
endmodule
```

- 仿真图象

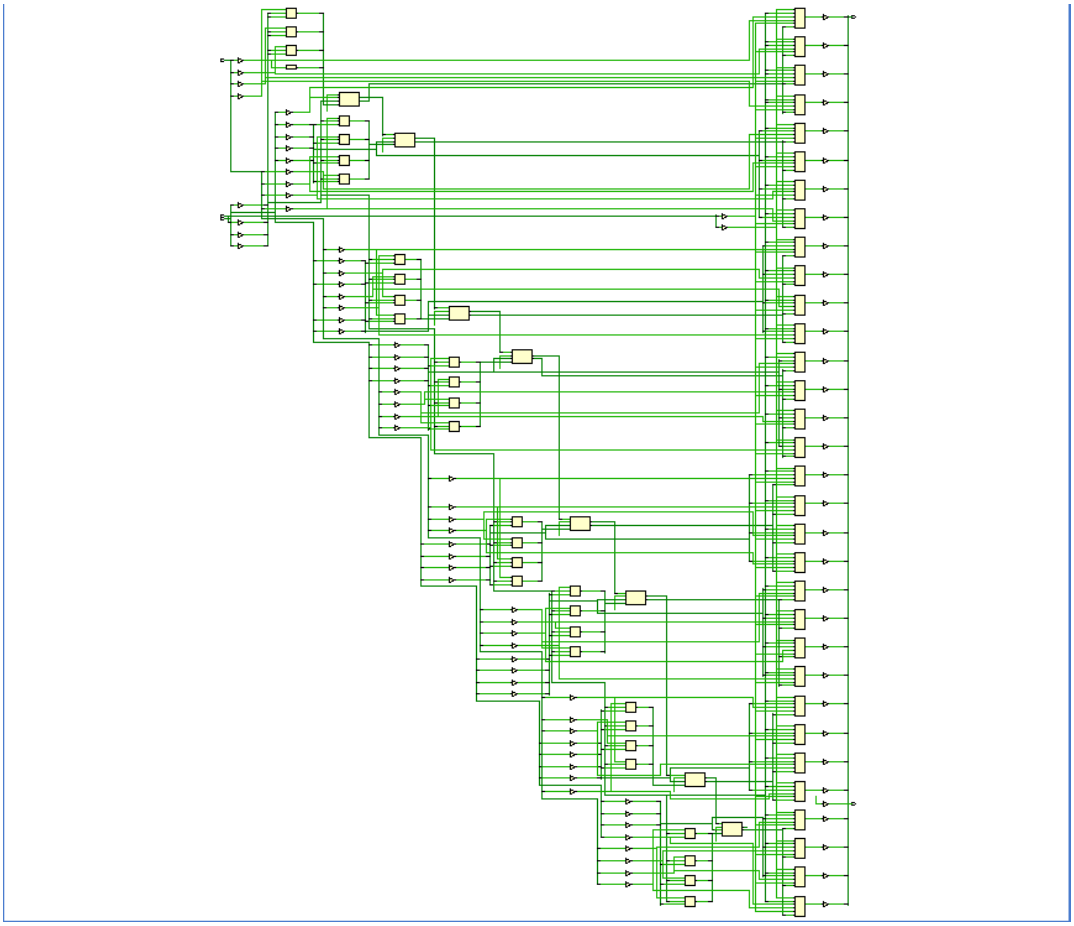


【二：性能报告】

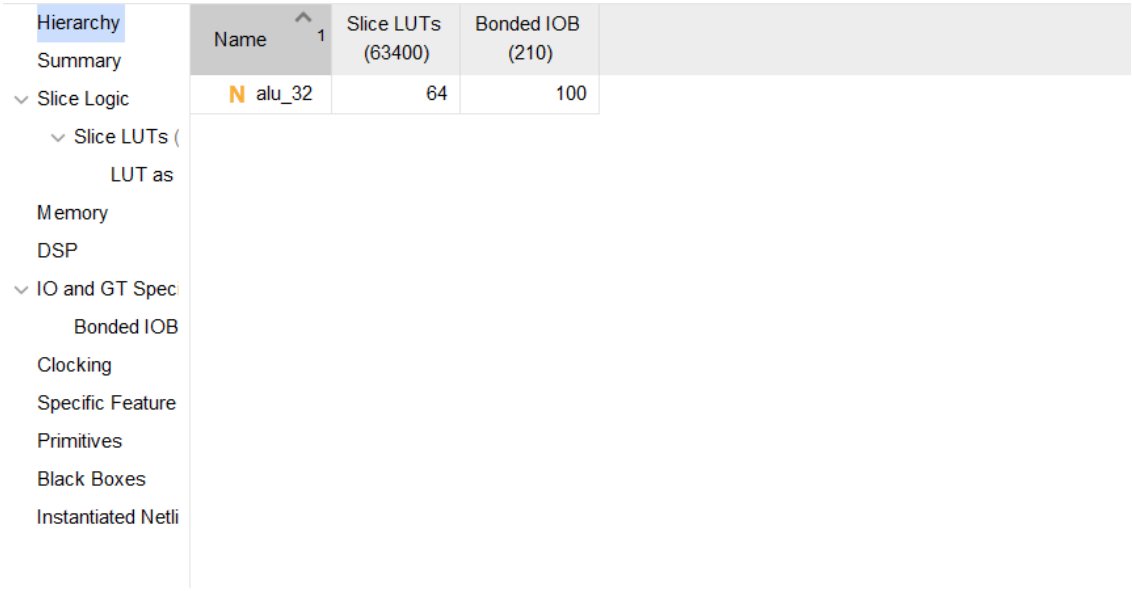
- 生成电路
  - RTL电路：


a[31:0]
b[31:0]
y0_i
I0[31:0]
I1[31:0]
+
O[31:0]
RTL_ADD
y0_i__0
-
RTL_SUB
y0_i__1
RTL_AND
y0_i__2
RTL_OR
y0_i__3
RTL_XOR
y_i
S=3'b000 I0[31:0]
S=3'b001 I1[31:0]
S=3'b010 I2[31:0]
S=3'b011 I3[31:0]
S=3'b100 I4[31:0]
S=default I5[31:0]
S[2:0]
RTL_MUX
y[31:0]
f[2:0]
z_i
A[2:0]
O
RTL_ROM
z_reg
D
G
Q
RTL_LATCH
z

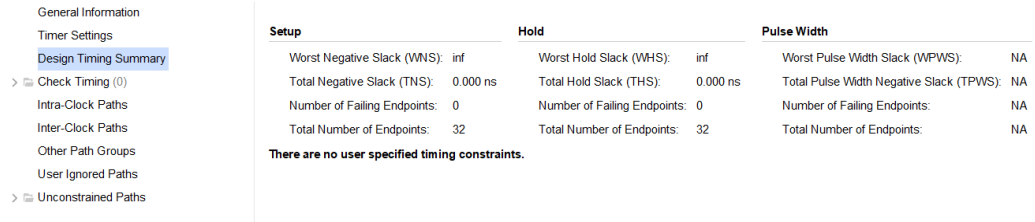
- 综合电路：



- 资源使用情况

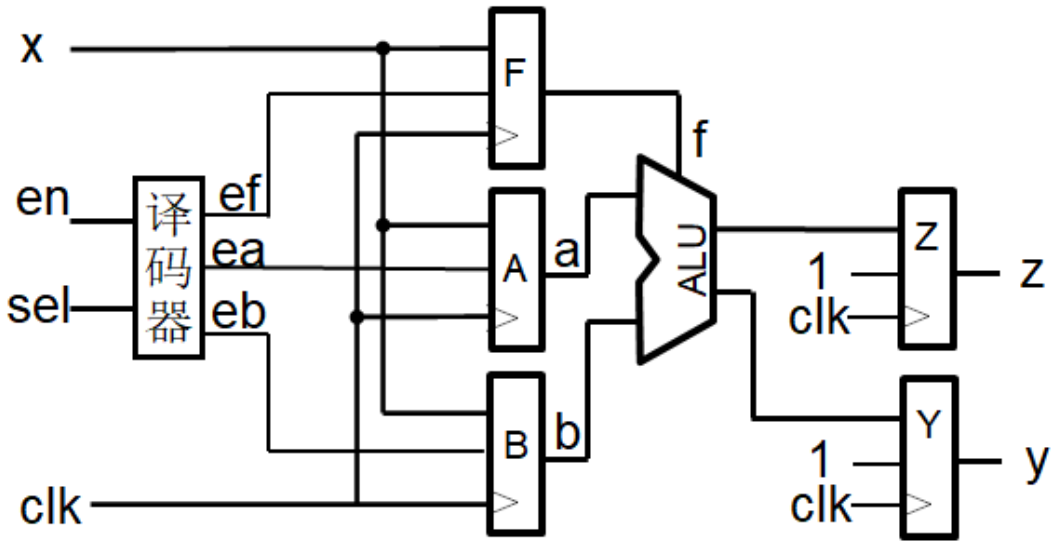


- 综合电路性能(由于32位ALU未烧在板子上故不做时间性能的特定分析)



【三：6位ALU】

- 数据通路



- 状态图（真值表）
-

en	sel	ea	eb	ef
1	00	1	0	0
1	01	0	1	0
1	10	0	0	1
0	xx	0	0	0

- 端口分配

端口	外设
clk	100MHz
en	button
sel	sw[7:6]
x	sw[5:0]
y	led[5:0]
z	led[7]

- 设计文件

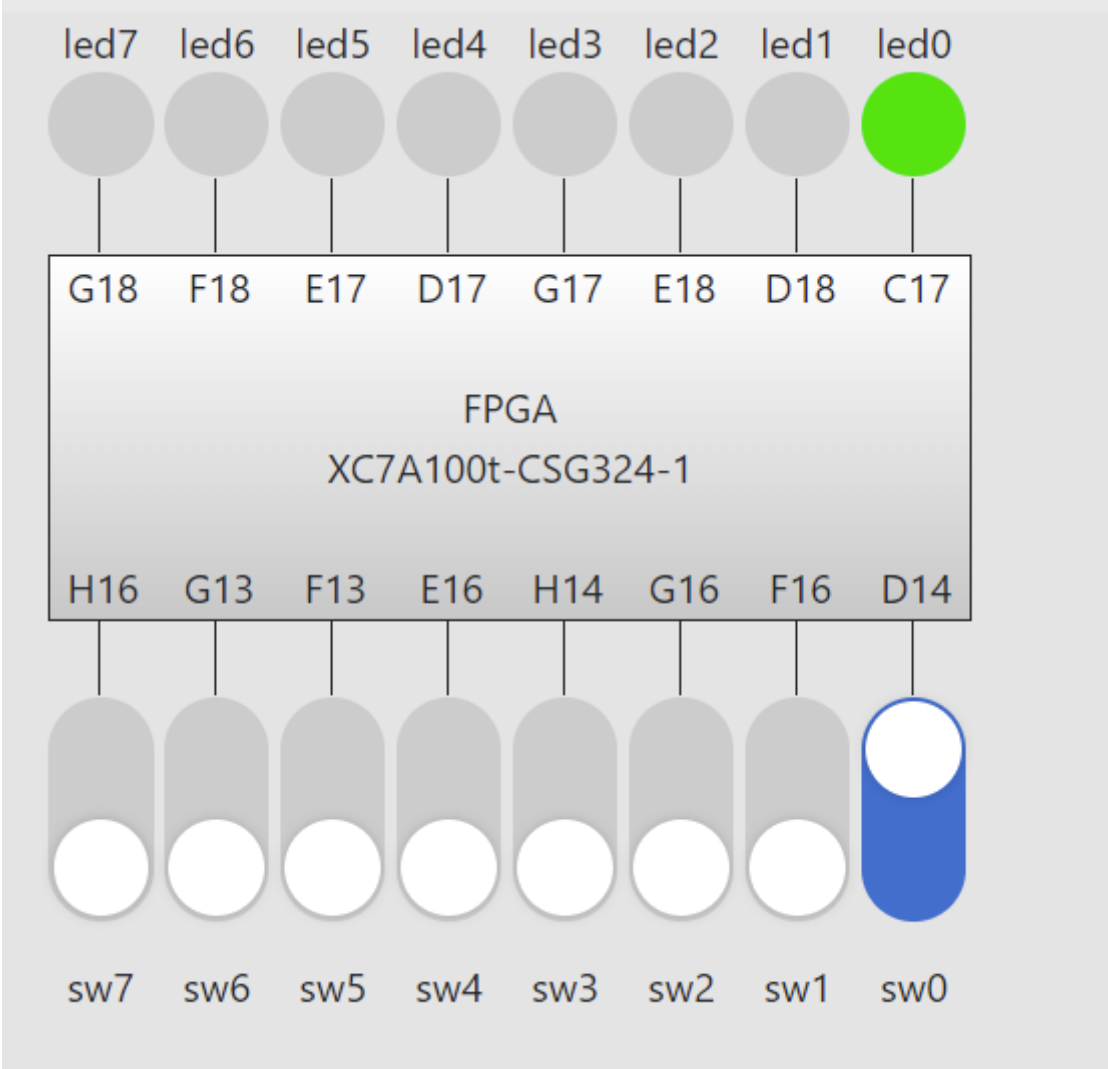
```
module alu_6(
    input clk,
    input en,
    input [1:0] sel,
    input [5:0] x,
    output reg [5:0] y,
    output reg z
    );
    reg [2:0] f;
    reg [5:0] a, b;
    always@(posedge clk)
    begin
        if(en)
        case(sel)
        2'b00:a <= x;
        2'b01:b <= x;
        2'b10:f <= x[2:0];
        endcase
    end

    always@(posedge clk)
    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:
            begin
                y <= 0;
                z <= 1;
            end
        endcase
    end
endmodule
```

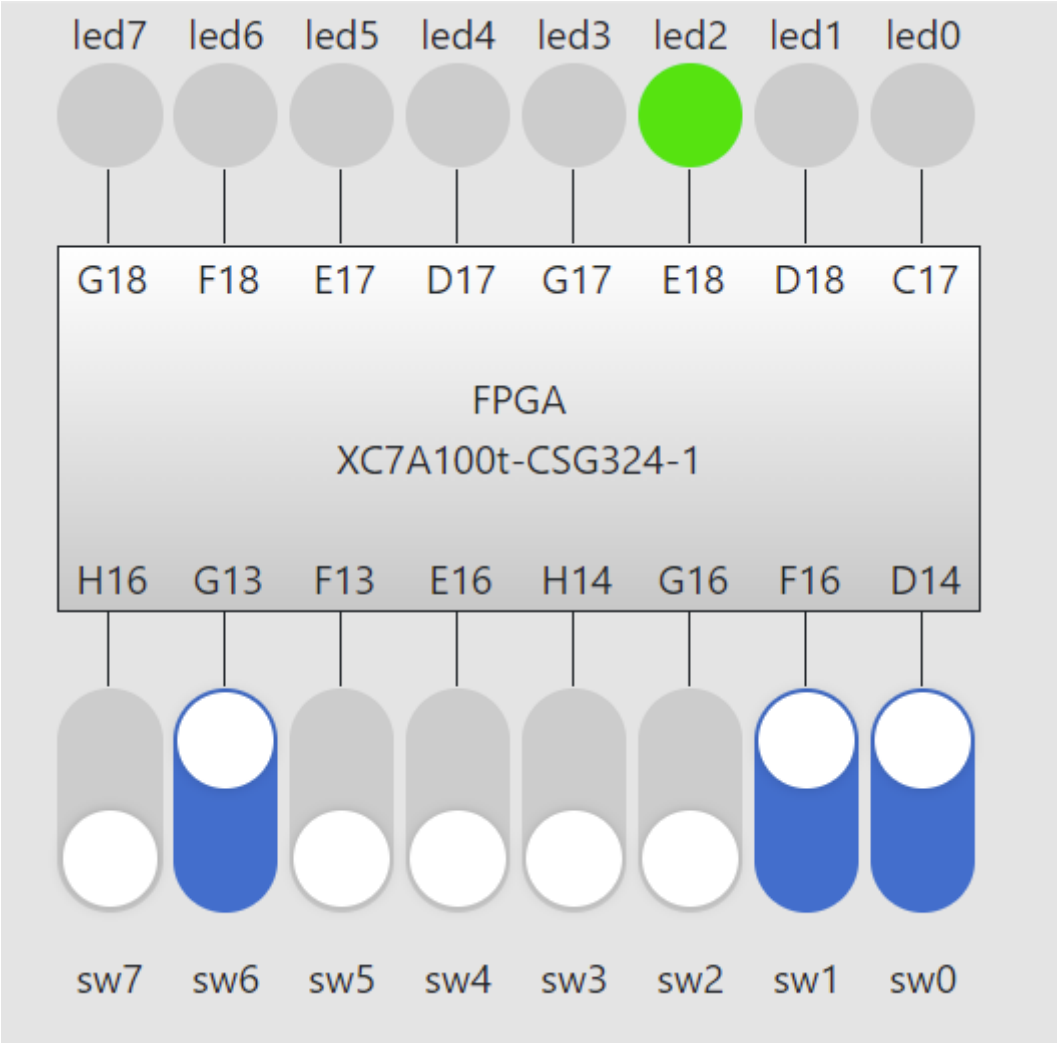
- 约束文件

```
## Clock signal
set_property -dict {PACKAGE_PIN E3 IOSTANDARD LVCMOS33} [get_ports clk]
#create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports {CLK100MHZ}];


## FPGAOL LED (signle-digit-SEGPLAY)

set_property -dict {PACKAGE_PIN C17 IOSTANDARD LVCMOS33} [get_ports {y[0]}]
set_property -dict {PACKAGE_PIN D18 IOSTANDARD LVCMOS33} [get_ports {y[1]}]
set_property -dict {PACKAGE_PIN E18 IOSTANDARD LVCMOS33} [get_ports {y[2]}]
set_property -dict {PACKAGE_PIN G17 IOSTANDARD LVCMOS33} [get_ports {y[3]}]
set_property -dict {PACKAGE_PIN D17 IOSTANDARD LVCMOS33} [get_ports {y[4]}]
set_property -dict {PACKAGE_PIN E17 IOSTANDARD LVCMOS33} [get_ports {y[5]}]
set_property -dict { PACKAGE_PIN F18   IOSTANDARD LVCMOS33 } [get_ports { }];
set_property -dict {PACKAGE_PIN G18 IOSTANDARD LVCMOS33} [get_ports z]


## FPGAOL SWITCH

set_property -dict {PACKAGE_PIN D14 IOSTANDARD LVCMOS33} [get_ports {x[0]}]
set_property -dict {PACKAGE_PIN F16 IOSTANDARD LVCMOS33} [get_ports {x[1]}]
set_property -dict {PACKAGE_PIN G16 IOSTANDARD LVCMOS33} [get_ports {x[2]}]
set_property -dict {PACKAGE_PIN H14 IOSTANDARD LVCMOS33} [get_ports {x[3]}]
set_property -dict {PACKAGE_PIN E16 IOSTANDARD LVCMOS33} [get_ports {x[4]}]
set_property -dict {PACKAGE_PIN F13 IOSTANDARD LVCMOS33} [get_ports {x[5]}]
set_property -dict {PACKAGE_PIN G13 IOSTANDARD LVCMOS33} [get_ports {sel[0]}]
set_property -dict {PACKAGE_PIN H16 IOSTANDARD LVCMOS33} [get_ports {sel[1]}]
```

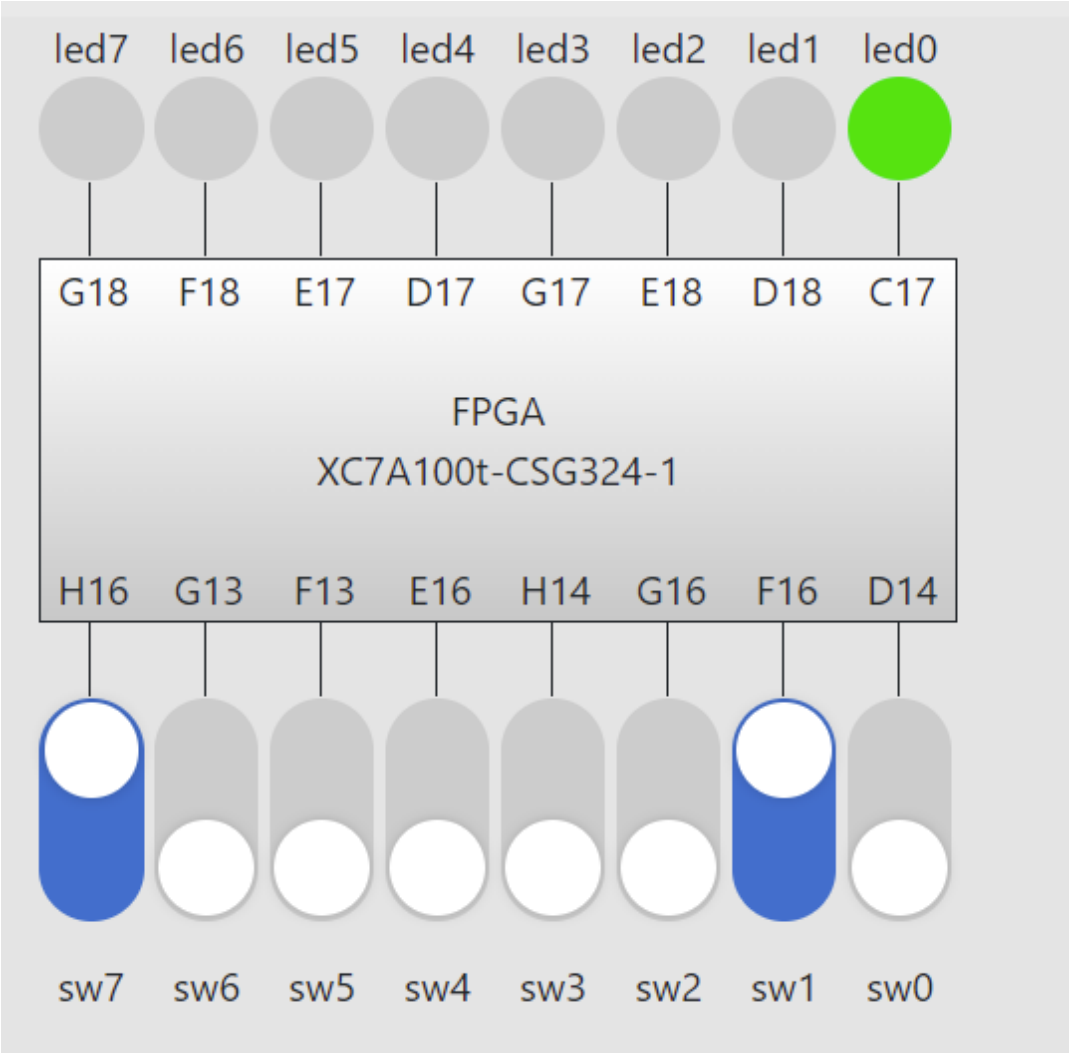
```
## FPGAOL BUTTON & SOFT_CLOCK

set_property -dict {PACKAGE_PIN B18 IOSTANDARD LVCMOS33} [get_ports en]
```

- FPGA运行结果（设a = 1, b = 3)

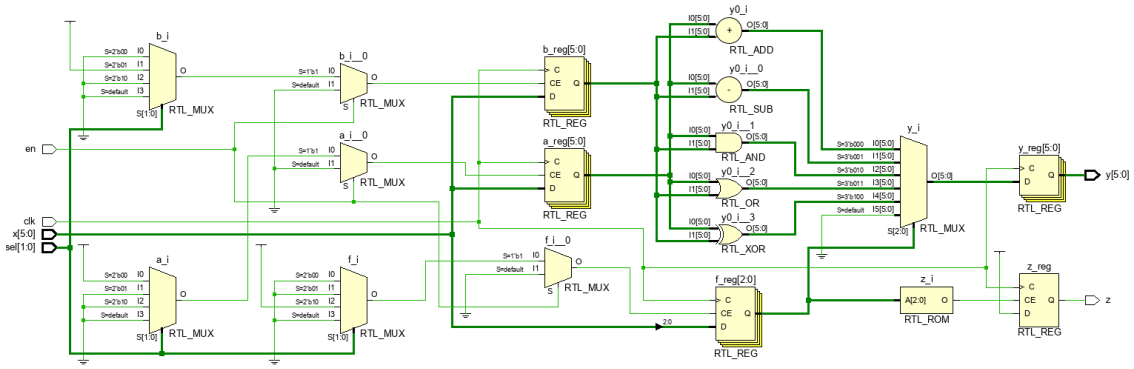




- 测试与功能（f赋值010）



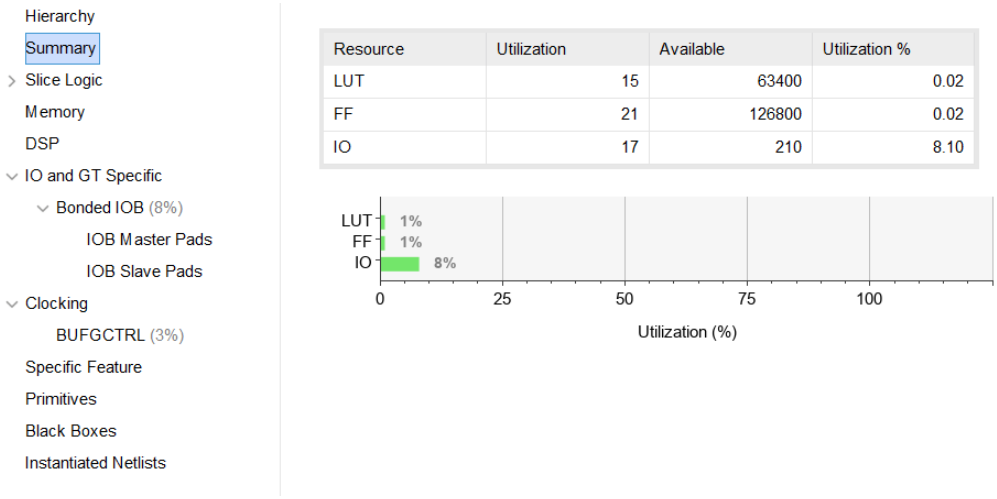
- RTL电路图



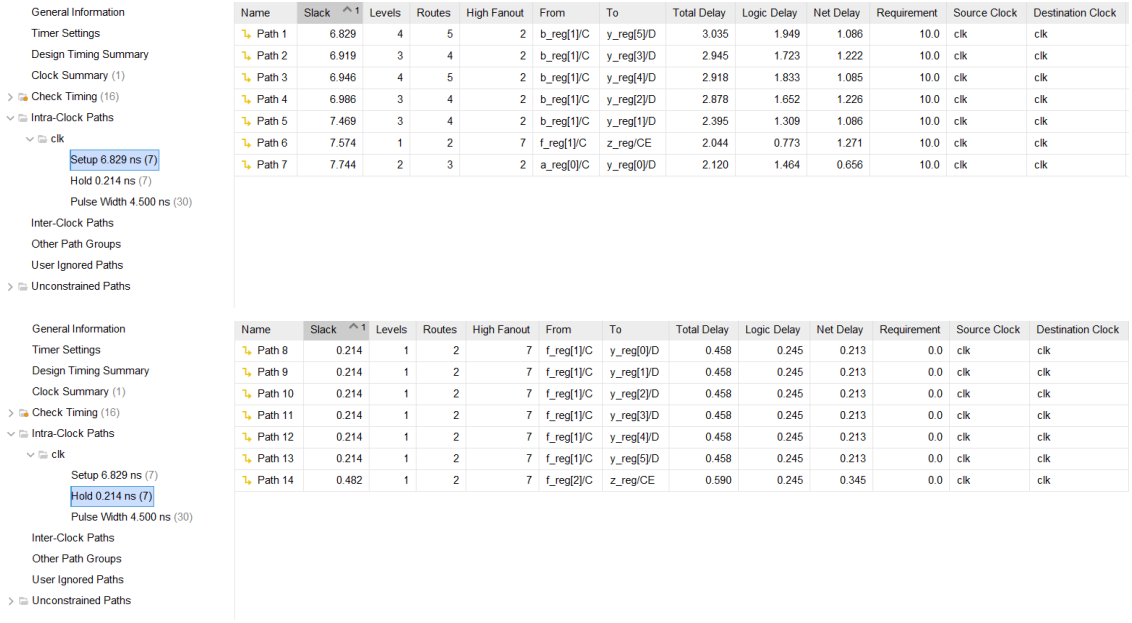
- 资源使用情况

Hierarchy
Summary
Slice Logic
Memory
DSP

Name	Slice LUTs (63400)	Slice Registers (126800)	Bonded IOB (210)	BUFGCTRL (32)
alu_6	15	21	17	1



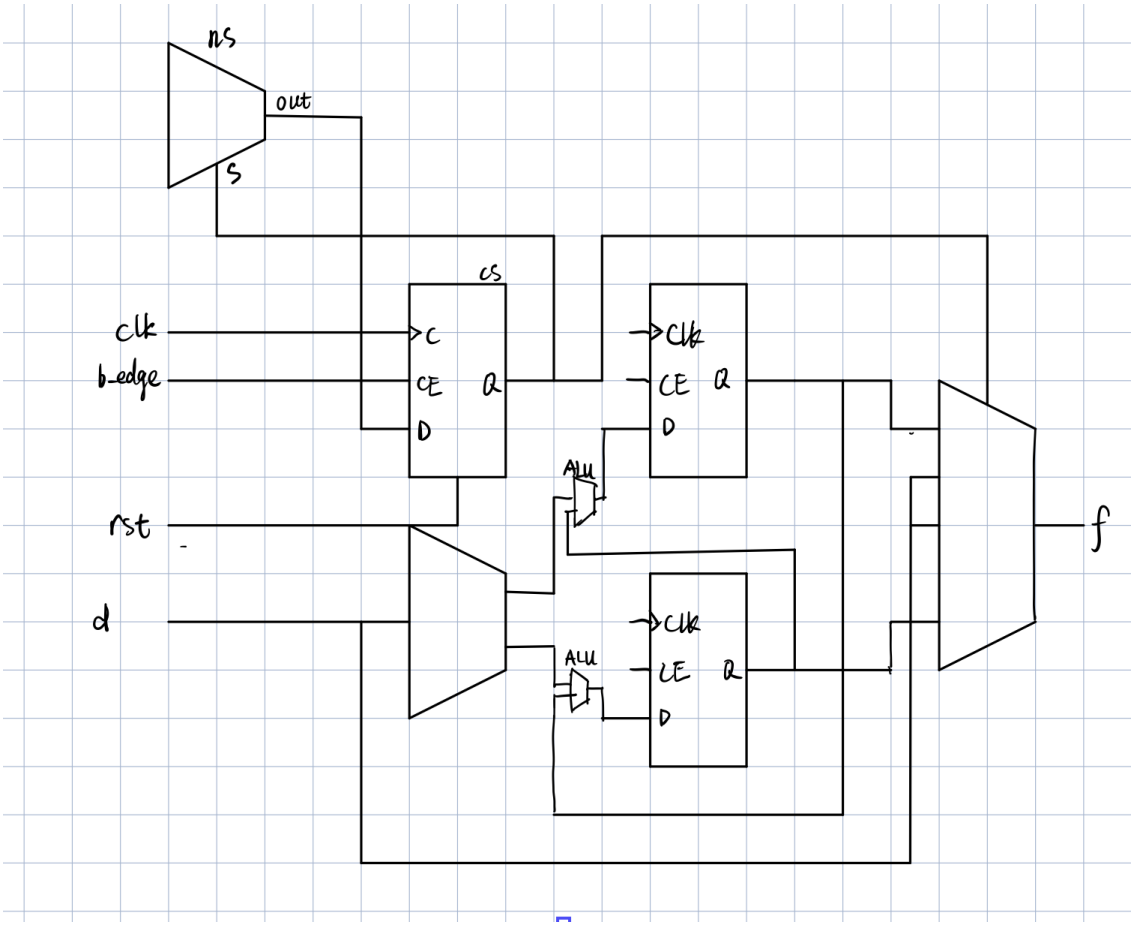
- 综合电路时间性能（



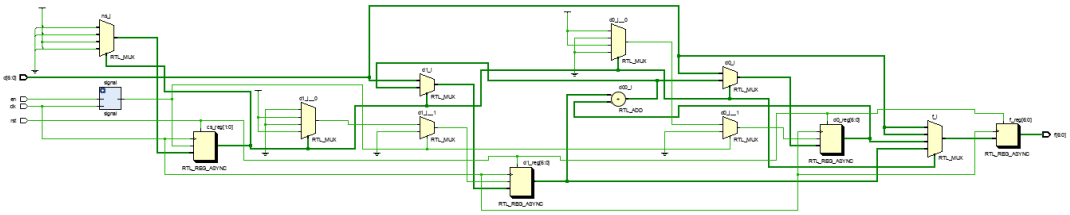
slack为正即满足setup time/hold time需求，high fanout为模块直接调用下级模块数量，其后则为各线路时间延迟。

【四：FLS】

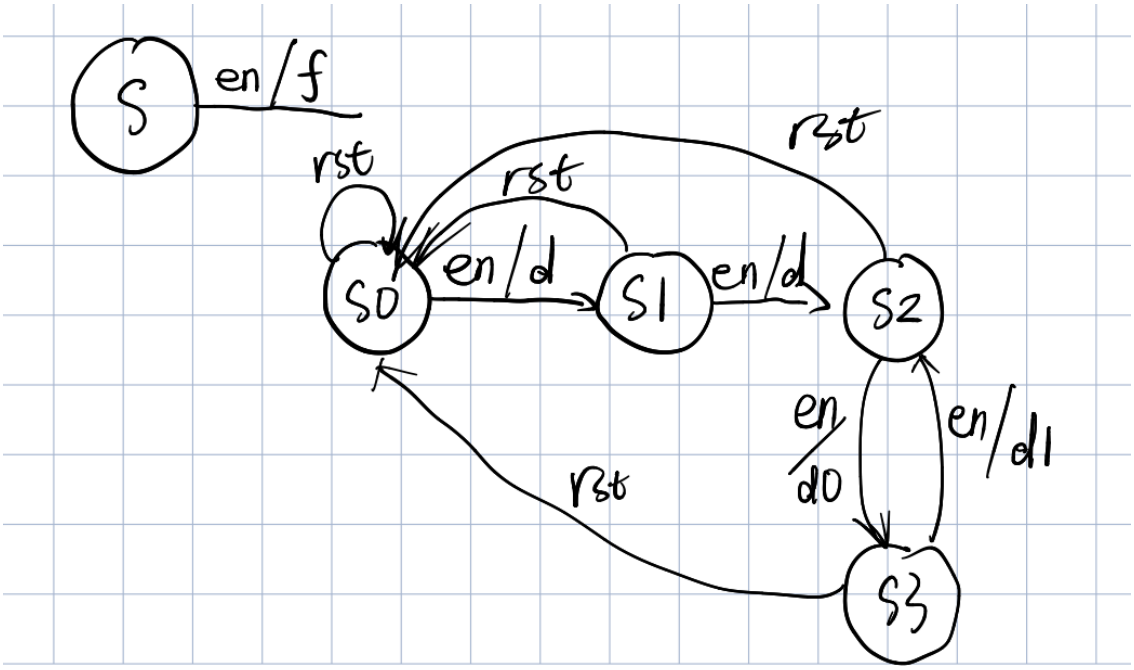
- 数据通路



与后续得到的RTL电路比较大致一致



- 状态图



- 设计文件

```
//信号模块用于取按钮边沿
module signal(
    input clk,button,
    output button_edge
);
    reg b1, b2;
    always@(posedge clk)
        b1 <= button;
    always @ (posedge clk)
        b2 <= b1;
    assign button_edge = b1 & (~b2);
endmodule

module fls(
    input clk, rst,
    input en,
    input [6:0] d,
    output reg [6:0] f
    );
    reg [6:0] d0, d1; //用于临时储存结果
    reg [1:0] cs;   //共四个状态，故只需两位
    reg [1:0] ns;
    wire b_edge;    //按钮边沿
    signal signal(clk, en, b_edge);

    //next state
    always @ (*)
    begin
        case(cs)
            2'b00: ns = 2'b01;
            2'b01: ns = 2'b10;
            2'b10: ns = 2'b11;//转入状态二后即进行序列输出，在两个状态间来回切换
            2'b11: ns = 2'b10;
        endcase
    end

    //how current state change
    always @ (posedge clk or posedge rst)
        if(rst)
            cs <= 2'b00;
        else if(b_edge)
            cs <= ns;

    //output
    always @ (posedge clk or posedge rst)
    begin
        if(rst)
        begin
            d0 <= 0;
            d1 <= 0;
        end

        else if(b_edge)
            case(cs)
                2'b00://状态零，赋值给f0
```

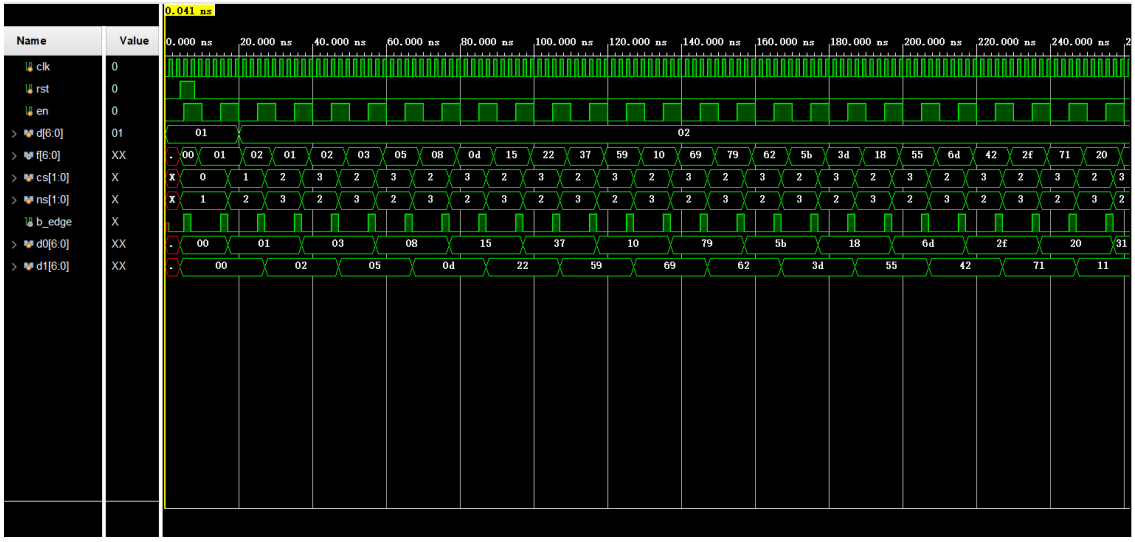
```
                begin
                    d0 <= d;
                end
                2'b01://状态一，赋值给f1
                begin
                    d1 <= d;
                end
                2'b10://状态二，此时d0排序靠后，累加至d0
                begin
                    d0 <= d1 + d0;
                end
                2'b11://状态三，此时d1排序靠后，累加至d1
                begin
                    d1 <= d1 + d0;
                end
                endcase
    end

    always @ (posedge clk or posedge rst)
    begin
    if(rst)
        f <= 0;
    else
        case(cs)
            2'b00:
            begin
                f <= d;
            end
            2'b01:
            begin
                f <= d;
            end
            2'b10:
            begin
                f <= d0;
            end
            2'b11:
            begin
                f <= d1;
            end
        endcase
    end
endmodule
```

- 仿真文件

```
module sim1(

    );
    reg clk, rst, en;
    reg [6:0] d;
    wire [6:0] f;
    fls fls(clk, rst, en, d, f);
    initial
    begin
    rst = 0;
    #4 rst = 1;
```

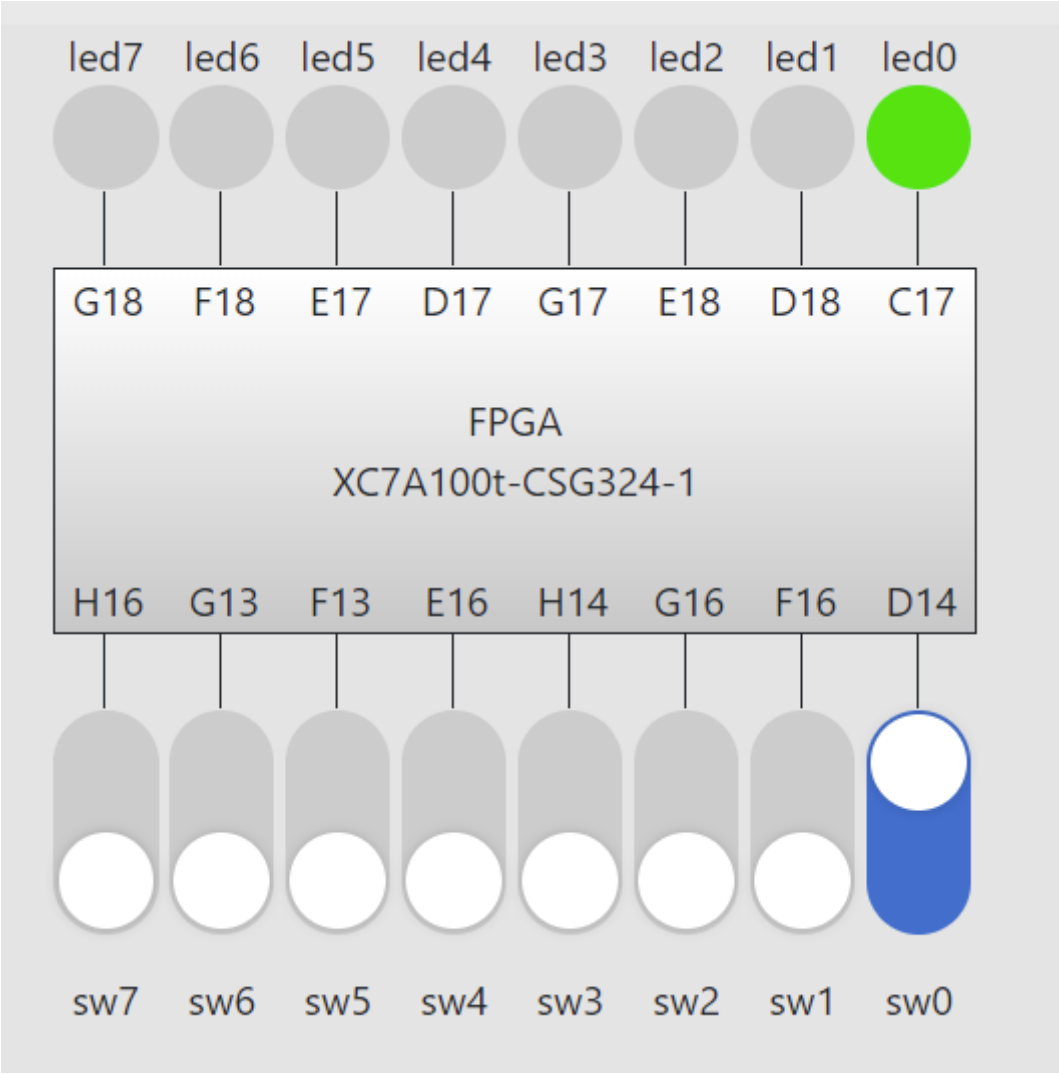
```
    #4 rst = 0;
    end
    initial clk = 0;
    always #1 clk = ~clk;
    initial en = 0;
    always #5 en = ~en;
    initial
    begin
        d = 1;#20 d = 2;
        end
endmodule
```

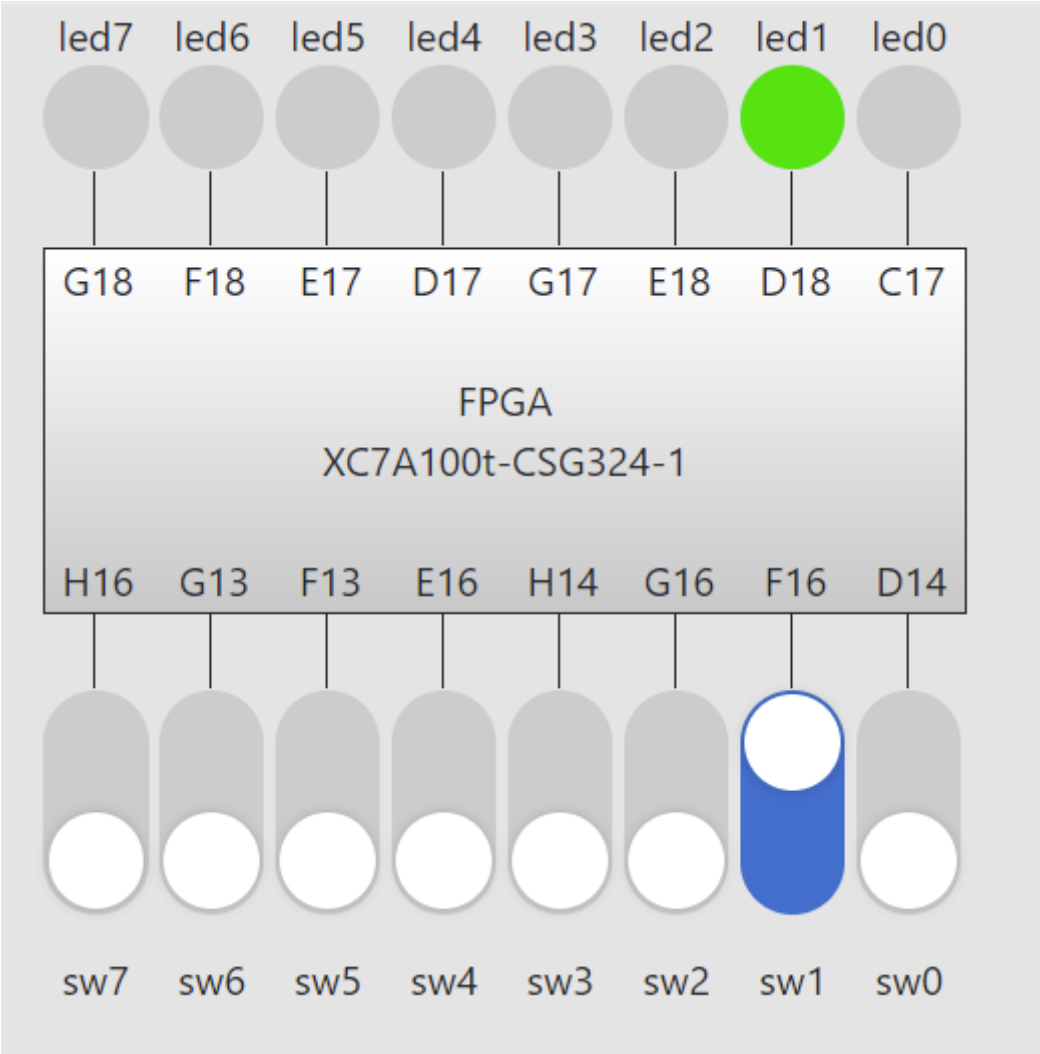
- 仿真结果



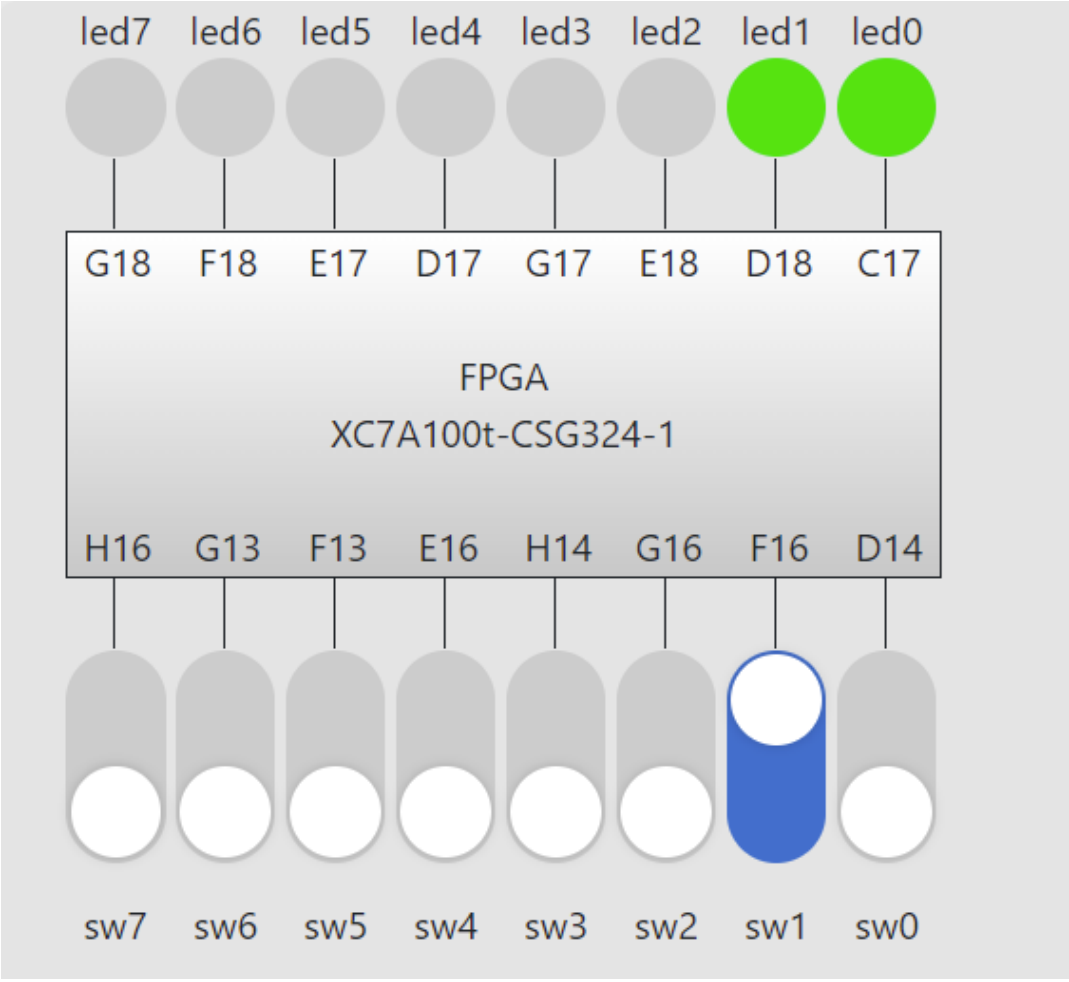
可见在输入初始的两项之后f按照斐波那契序列输出。

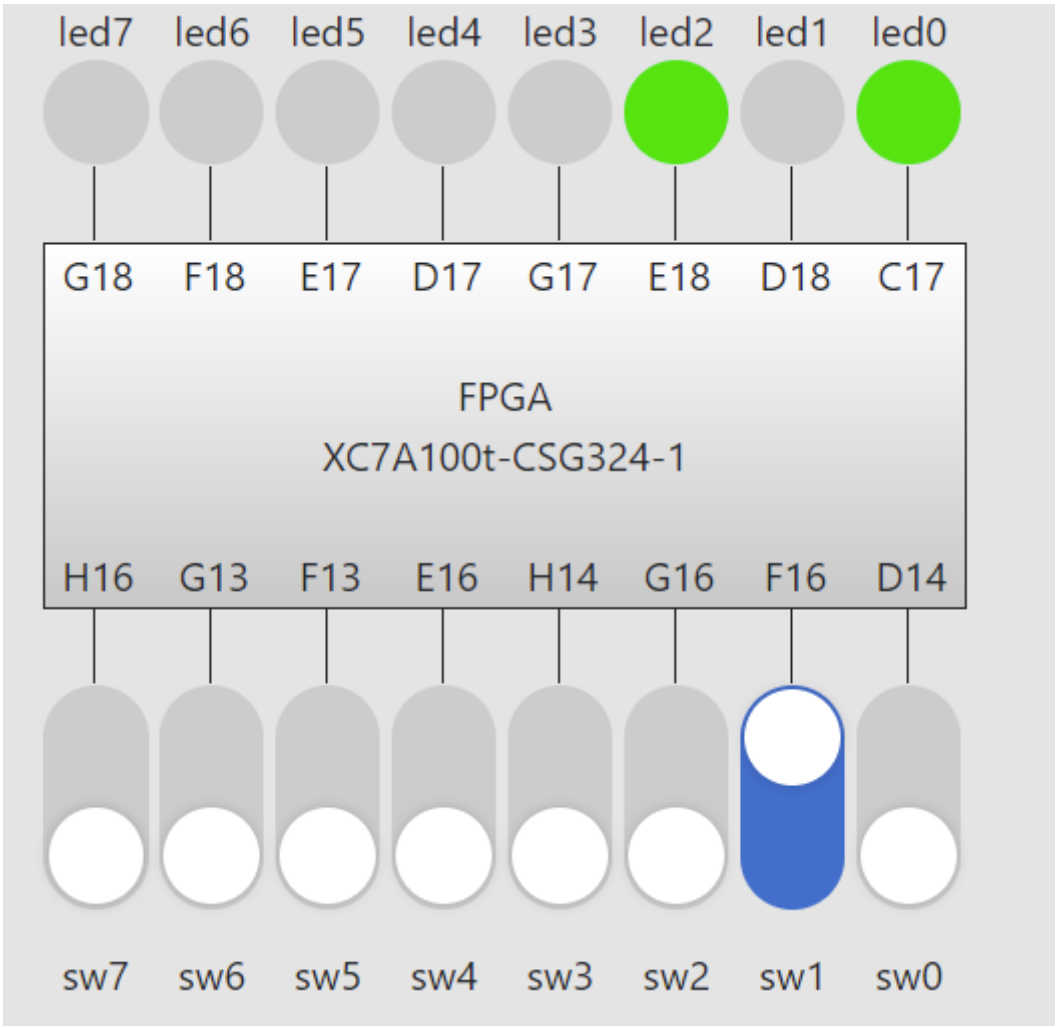
- FPGA测试结果
  - 输入a = 1, b = 2


led7
led6
led5
led4
led3
led2
led1
led0
G18
F18
E17
D17
G17
E18
D18
C17
FPGA
XC7A100t-CSG324-1
H16
G13
F13
E16
H14
G16
F16
D14
sw7
sw6
sw5
sw4
sw3
sw2
sw1
sw0



- 此后每一次均输出斐波那契的下一项


led7 led6 led5 led4 led3 led2 led1 led0
G18 F18 E17 D17 G17 E18 D18 C17
FPGA
XC7A100t-CSG324-1
H16 G13 F13 E16 H14 G16 F16 D14
sw7 sw6 sw5 sw4 sw3 sw2 sw1 sw0


led7 led6 led5 led4 led3 led2 led1 led0
G18 F18 E17 D17 G17 E18 D18 C17
FPGA
XC7A100t-CSG324-1
H16 G13 F13 E16 H14 G16 F16 D14
sw7 sw6 sw5 sw4 sw3 sw2 sw1 sw0

【总结与思考】

- 实验总结
  - 总体来说难度不高，主要用于复习上学期所学的模电实验内容——尤其是三段式有限状态机，以及学习如何对当前电路性能进行查看和评测；
- 实验建议（吐槽）
  - 作为一个复习模电的实验感觉没啥问题，确实花了蛮久复习代码怎么写（比如取边沿），但是画数据通路略有些折磨。