计算机组成原理 实验一报告

姓名：曾郅琛　学号：PB20071431　实验日期：2022-3-24

一、实验题目：

Lab01 运算器及其应用

二、实验目的：

设计算术逻辑运算单元（ALU），掌握数据通路和控制器设计方法

实现 32 位操作数 ALU 加、减、与、或、异或功能，利用前述的 32 位 ALU 模块设计 6 位操作数 ALU

设计计算斐波拉契数列 FLS，完成逻辑设计、仿真和下载测试

三、实验平台：

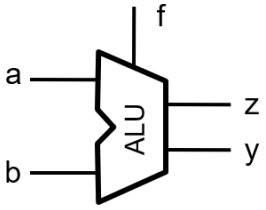
Vivado

四、实验过程：

1. ALU 逻辑设计

```
module alu #(parameter WIDTH=32)
    (
        input [WIDTH-1:0] a,b,
        input [2:0]f,
        output reg [WIDTH-1:0] y,
        output reg z
    );
    localparam ADD=0, SUB=1, AND=2, OR=3, XOR=4;
    always @(*)
    begin
        case(f)
            ADD: y = a+b;
            SUB: y = a-b;
            AND: y = a&b;
            OR:  y = a|b;
            XOR: y = a^b;
            default:
                begin y = 0 ;
                      z = 1;
                end
       endcase
       if(y) z=0;
       else  z=1;
    end
endmodule
```

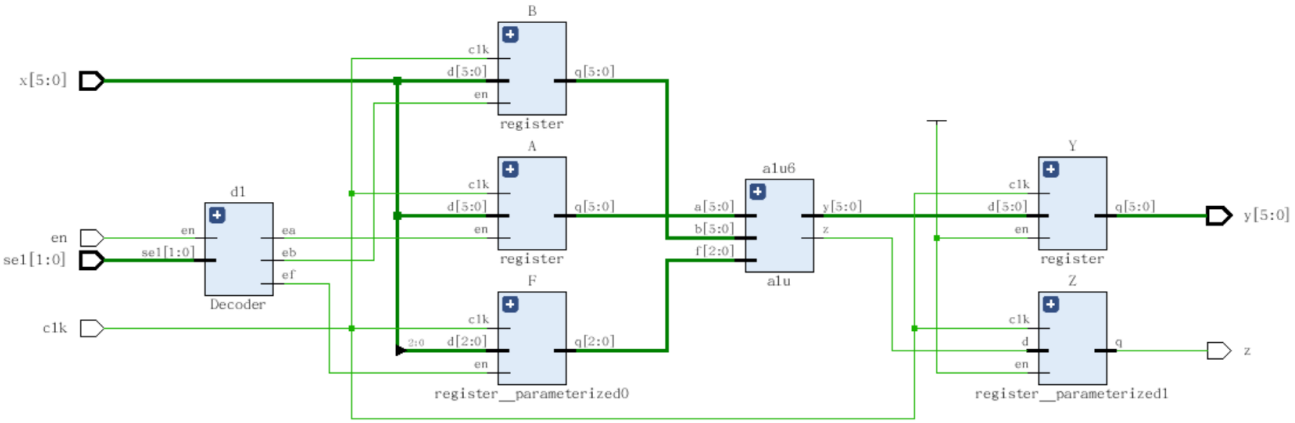
在 ALU 实现中，f 实现操作功能，y 作为运算结果输出，按照模块功能表执行功能.



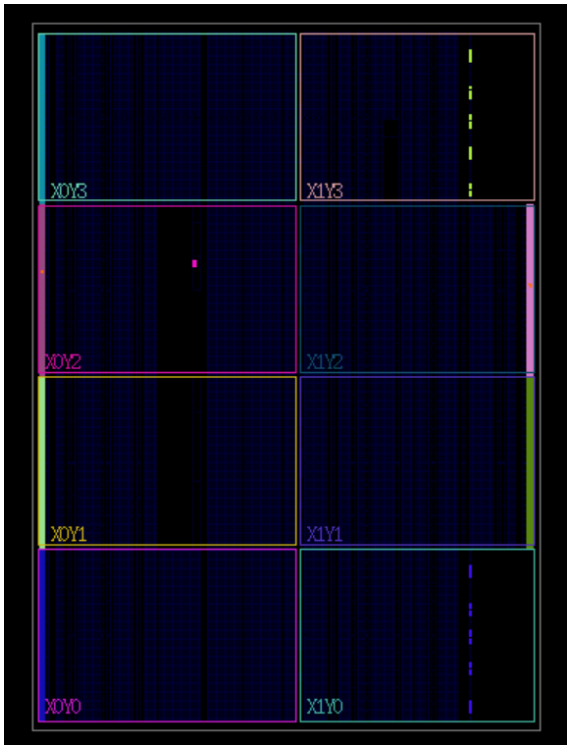
ALU 模块功能表

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

RTL 电路：



综合电路：



资源使用情况

综合电路：

Utilization

Hierarchy

Hierarchy
Summary
Slice Logic
Slice LUTs (<1%)
LUT as Logic (<1%)
Slice Registers (<1%)
Register as Latch (
Register as Flip Fl
Memory
DSP

Name	Slice LUTs (63400)	Slice Registers (126800)	Bonded IOB (210)	BUFGCTRL (32)
fls	19	29	17	1
alu_7 (alu)	0	0	0	0
D1 (diff)	1	1	0	0
D2 (diff_0)	3	1	0	0
r1 (register1)	0	7	0	0
r2 (register1_1)	14	7	0	0

ALU 模块综合电路性能及时间性能报告：

Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	7.823 ns	Worst Hold Slack (WHS):	0.159 ns	Worst Pulse Width Slack (WPWS):	4.500 ns
Total Negative Slack (TNS):	0.000 ns	Total Hold Slack (THS):	0.000 ns	Total Pulse Width Negative Slack (TPWS):	0.000 ns
Number of Failing Endpoints:	0	Number of Failing Endpoints:	0	Number of Failing Endpoints:	0
Total Number of Endpoints:	25	Total Number of Endpoints:	25	Total Number of Endpoints:	20

All user specified timing constraints are met.

Timing

General Information
Timer Settings
Design Timing Summary
Clock Summary (1)
Check Timing (55)
Intra-Clock Paths
sys_clk_pin
Setup 7.823 ns (10)
Hold 0.159 ns (10)
Pulse Width 4.500 ns (30
Inter-Clock Paths
Other Path Groups
User Ignored Paths
Unconstrained Paths

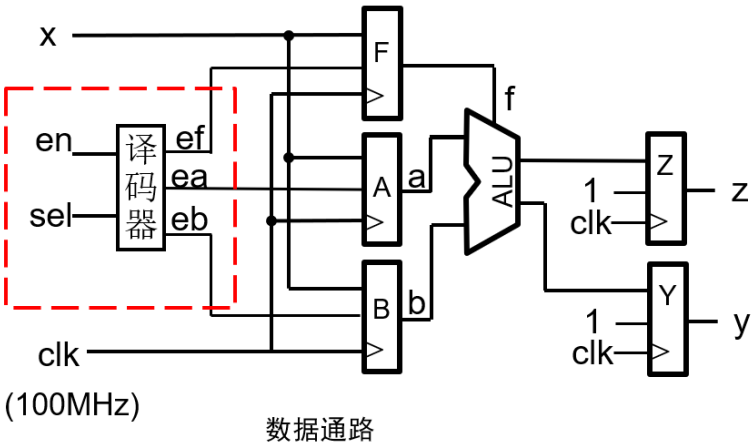
Intra-Clock Paths - sys_clk_pin - Setup

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement
Path 1	7.823	1	2	14	D1/b_reg/C	r1/data_reg[0]/CE	1.795	0.773	1.022	10.0
Path 2	7.823	1	2	14	D1/b_reg/C	r1/data_reg[1]/CE	1.795	0.773	1.022	10.0
Path 3	7.823	1	2	14	D1/b_reg/C	r1/data_reg[2]/CE	1.795	0.773	1.022	10.0
Path 4	7.823	1	2	14	D1/b_reg/C	r1/data_reg[3]/CE	1.795	0.773	1.022	10.0
Path 5	7.823	1	2	14	D1/b_reg/C	r1/data_reg[4]/CE	1.795	0.773	1.022	10.0
Path 6	7.823	1	2	14	D1/b_reg/C	r1/data_reg[5]/CE	1.795	0.773	1.022	10.0
Path 7	7.823	1	2	14	D1/b_reg/C	r1/data_reg[6]/CE	1.795	0.773	1.022	10.0
Path 8	7.823	1	2	14	D1/b_reg/C	r2/data_reg[0]/CE	1.795	0.773	1.022	10.0
Path 9	7.823	1	2	14	D1/b_reg/C	r2/data_reg[1]/CE	1.795	0.773	1.022	10.0
Path 10	7.823	1	2	14	D1/b_reg/C	r2/data_reg[2]/CE	1.795	0.773	1.022	10.0

2. 6 位操作数 ALU 逻辑设计

译码器真值表

en	sel	ena	enb	ef
1	00	1	0	0
1	01	0	1	0
1	10	0	0	1
1	11	0	0	0
0	xx	0	0	0



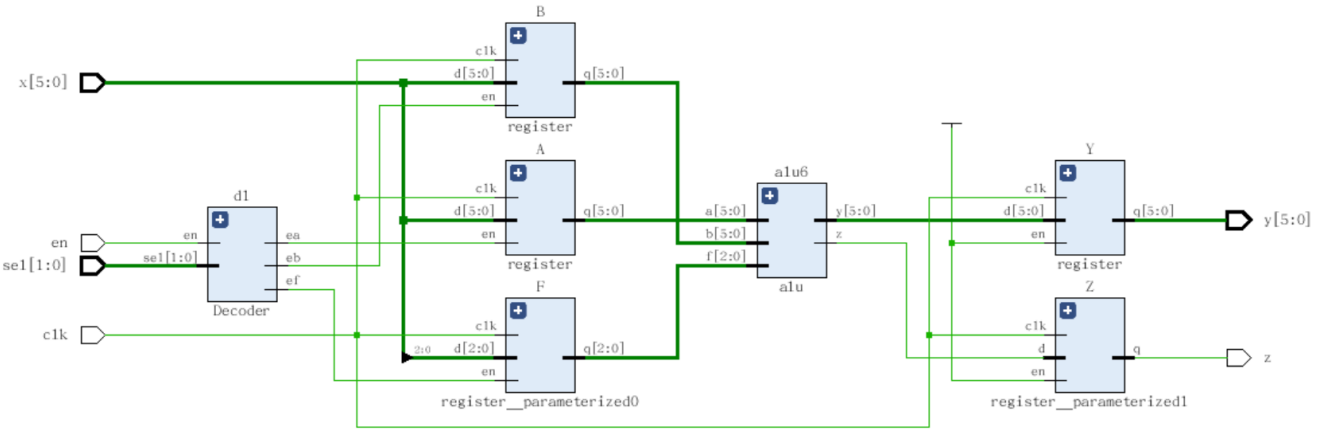
数据通路

```
`timescale 1ns / 1ps

module  register
    #(parameter WIDTH=32,
    RST_VALUE=0)
    (input  clk, en,
    input  [WIDTH-1 : 0]  d,
    output reg  [WIDTH-1 : 0] q);
always @(posedge clk)
    if (en)
        q <= d;
endmodule

module Decoder(
    input [1:0] sel,
    input en,
    output wire ef,ea,eb//ef,ea,eb
);

assign ea = ~sel[1]& ~sel[0]  & en;//00
assign eb = ~sel[1]& sel[0]  & en;//01
assign ef = sel[1]& ~sel[0]  & en;//10

endmodule
```

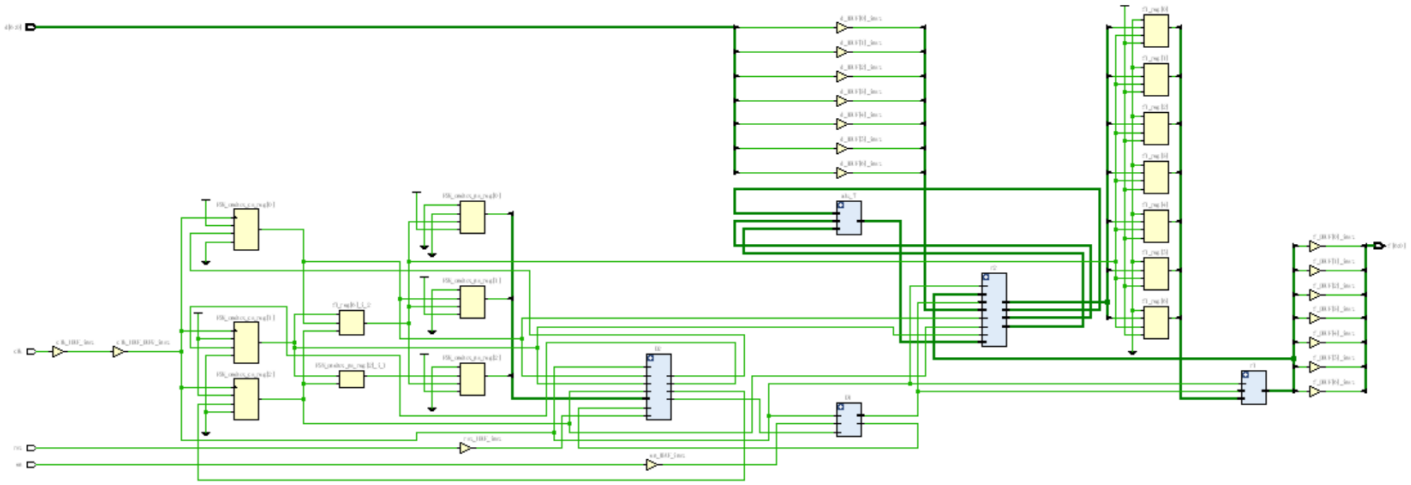
```
module alu_6
(
        input en,clk,
        input [1:0]sel,
        input [5:0]x,
        output [5:0] y,
        output  z
);
wire ef,ea,eb;
wire [5:0] a,b;
wire [2:0] f;
wire [5:0] y_in;
wire z_in;

Decoder d1(.en(en),.sel(sel),.ea(ea),.eb(eb),.ef(ef));

register #(6,0) A(.clk(clk),.en(ea),.d(x[5:0]),.q(a[5:0]));
register #(6,0) B(.clk(clk),.en(eb),.d(x[5:0]),.q(b[5:0]));
register #(3,0) F(.clk(clk),.en(ef),.d(x[2:0]),.q(f[2:0]));

alu #(6) alu6 (.a(a),.b(b),.f(f),.y(y_in),.z(z_in));
register #(6,0) Y(.clk(clk),.en(1),.d(y_in[5:0]),.q(y[5:0]));
register #(1,0) Z(.clk(clk),.en(1),.d(z_in),.q(z));
endmodule
```

设计译码器对 ef、ea、eb，选择 ALU 计算（按上图编写数据通路），操作数 a, b 和功能 f 复用开关输入 x[5:0]。通过 sel 和 en，生成译码电路，将开关输入 x[5:0]分时存入寄存器 F(x[2:0])，A(x[5:0])，B(x[5:0])。

RTL 电路图：



综合电路图：



Hierarchy

Name	Slice LUTs (63400)	Slice Registers (126800)	Bonded IOB (210)	BUFGCTRL (32)
fls	19	29	17	1
alu_7 (alu)	0	0	0	0
D1 (diff)	1	1	0	0
D2 (diff_0)	3	1	0	0
r1 (register1)	0	7	0	0
r2 (register1_1)	14	7	0	0

时间性能报告：

Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	7.823 ns	Worst Hold Slack (WHS):	0.159 ns	Worst Pulse Width Slack (WPWS):	4.500 ns
Total Negative Slack (TNS):	0.000 ns	Total Hold Slack (THS):	0.000 ns	Total Pulse Width Negative Slack (TPWS):	0.000 ns
Number of Failing Endpoints:	0	Number of Failing Endpoints:	0	Number of Failing Endpoints:	0
Total Number of Endpoints:	25	Total Number of Endpoints:	25	Total Number of Endpoints:	20

All user specified timing constraints are met.

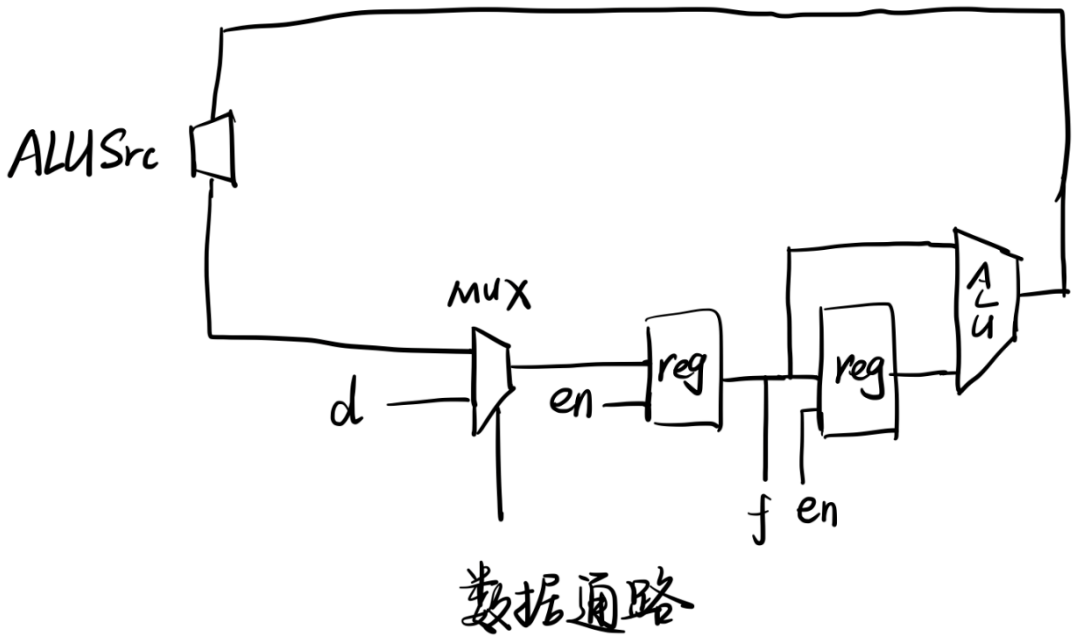
Intra-Clock Paths - sys_clk_pin - Setup

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement
Path 1	7.823	1	2	14	D1/b_reg/C	r1/data_reg[0]/CE	1.795	0.773	1.022	10.0
Path 2	7.823	1	2	14	D1/b_reg/C	r1/data_reg[1]/CE	1.795	0.773	1.022	10.0
Path 3	7.823	1	2	14	D1/b_reg/C	r1/data_reg[2]/CE	1.795	0.773	1.022	10.0
Path 4	7.823	1	2	14	D1/b_reg/C	r1/data_reg[3]/CE	1.795	0.773	1.022	10.0
Path 5	7.823	1	2	14	D1/b_reg/C	r1/data_reg[4]/CE	1.795	0.773	1.022	10.0
Path 6	7.823	1	2	14	D1/b_reg/C	r1/data_reg[5]/CE	1.795	0.773	1.022	10.0
Path 7	7.823	1	2	14	D1/b_reg/C	r1/data_reg[6]/CE	1.795	0.773	1.022	10.0
Path 8	7.823	1	2	14	D1/b_reg/C	r2/data_reg[0]/CE	1.795	0.773	1.022	10.0
Path 9	7.823	1	2	14	D1/b_reg/C	r2/data_reg[1]/CE	1.795	0.773	1.022	10.0
Path 10	7.823	1	2	14	D1/b_reg/C	r2/data_reg[2]/CE	1.795	0.773	1.022	10.0

Intra-Clock Paths - sys_clk_pin - Hold

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay
Path 11	0.159	0	1	3	r1/data_reg[0]/C	r2/data_reg[0]/D	0.295	0.147	0.148
Path 12	0.159	0	1	3	r1/data_reg[1]/C	r2/data_reg[1]/D	0.295	0.147	0.148
Path 13	0.159	0	1	3	r1/data_reg[2]/C	r2/data_reg[2]/D	0.295	0.147	0.148
Path 14	0.159	0	1	3	r1/data_reg[3]/C	r2/data_reg[3]/D	0.295	0.147	0.148
Path 15	0.159	0	1	3	r1/data_reg[4]/C	r2/data_reg[4]/D	0.295	0.147	0.148
Path 16	0.159	0	1	3	r1/data_reg[5]/C	r2/data_reg[5]/D	0.295	0.147	0.148
Path 17	0.159	0	1	3	r1/data_reg[6]/C	r2/data_reg[6]/D	0.295	0.147	0.148
Path 18	0.164	0	1	5	D1/b_reg/C	D2/b_reg/D	0.300	0.147	0.153
Path 19	0.292	1	2	4	D2/b_reg/C	FSM_onehot_cs_reg[0]/D	0.536	0.245	0.291
Path 20	0.292	1	2	4	D2/b_reg/C	FSM_onehot_cs_reg[1]/D	0.536	0.245	0.291

3. FLS 逻辑设计



```
module diff(
input a,clk,
output reg b
    );
    always@(posedge clk)
    b<=a;
endmodule
```

```
`timescale 1ns / 1ps

module  fls (
  input  clk, rst,         //时钟，复位（高电平有效）
  input en,             //输入输出使能
  input  [6:0]  d,       //输入数列初始项
  output [6:0]  f        //输出数列
);
parameter s0=2'b00,s1=2'b01,s2=2'b10;
reg [1:0]cs,ns;
reg [6:0]f1;
wire[6:0]f2,f3,f4;
wire zero;
wire en2,en3,e;
diff D1(en,clk,en2);      //en2=en
diff D2(en2,clk,en3);    //en3=en2
assign e=en2&~en3;    //当en2=1，en3=0时赋值为1
alu#(7) alu_7(f2,f3,3'b000,f4,zero);
register1#(7) r1(e,clk,f1,f2);
register1#(7) r2(e,clk,f2,f3);
```

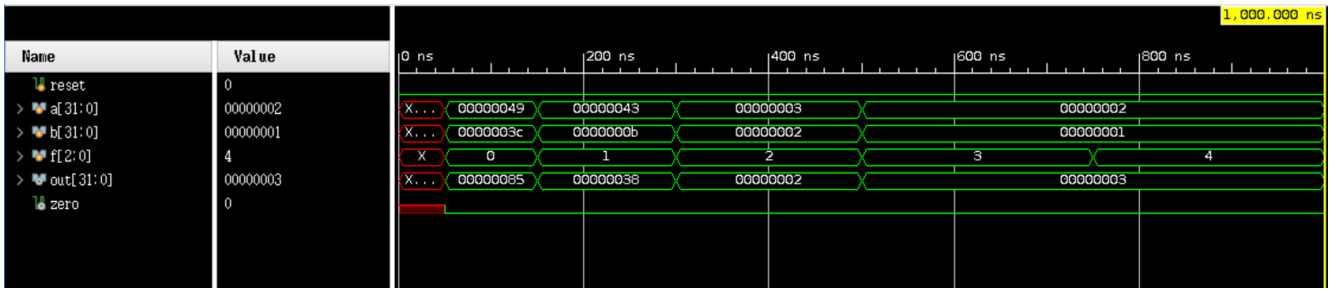
```
always@(posedge clk)
begin
if(rst)cs<=s0;
else if(e)cs<=ns;
else cs<=cs;
end
always@(*)
begin
case(cs)
2'b00:ns=s1;
2'b01:ns=s2;
2'b10:ns=s2;
default:;
endcase
end
always@(*)
case(cs)
2'b00:begin f1=d;end
2'b01:begin f1=d;end
2'b10:begin f1=f4;end
default:;
endcase
assign f=f2;
endmodule
```

五、实验结果：

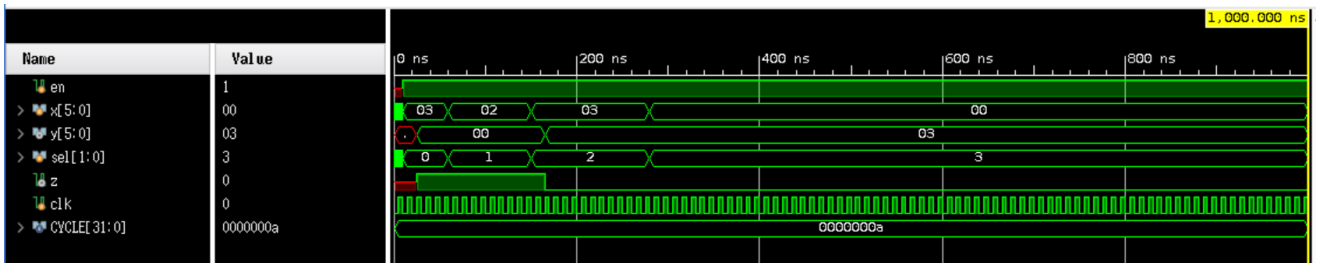
波形仿真：

1. ALU 仿真运算：

分别对 000、001、010、011、100 五个操作功能进行运算



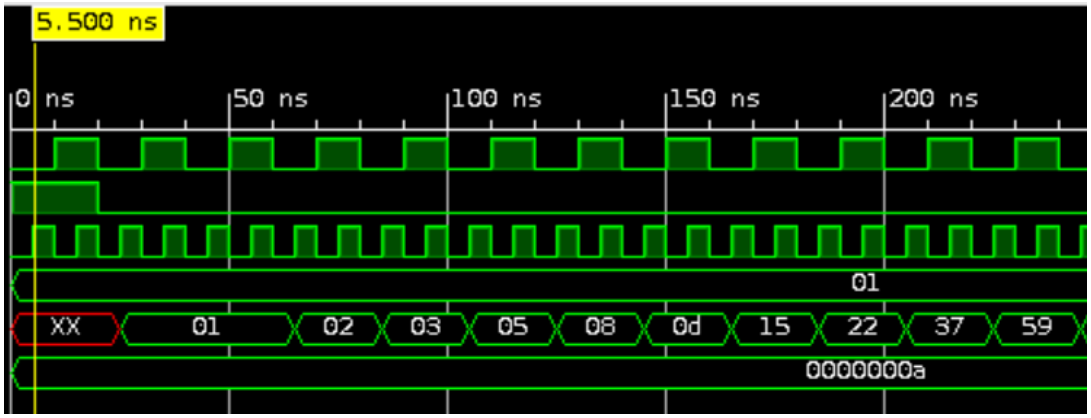
2. 6 位操作数 ALU 仿真波形



3. FLS 波形仿真：

```
module addtest3;
reg en;
reg rst=1;
reg clk;
reg [6:0] d=1;
wire [6:0] f;

// 时钟周期和个数

parameter CYCLE = 10;
initial  begin
  clk = 0;
forever
     #(CYCLE/2)  clk = ~ clk;
end

initial  begin
  en = 0;
forever
     #(CYCLE)  en = ~ en;
end

```

```
23      initial begin
24
25      #20
26      rst=0;
27      d=1;
28      end
29       fls sim3(
30      .en(en),
31      .rst(rst),
32      .clk(clk),
33      .d(d),
34      .f(f)
35          );
36      endmodule
```

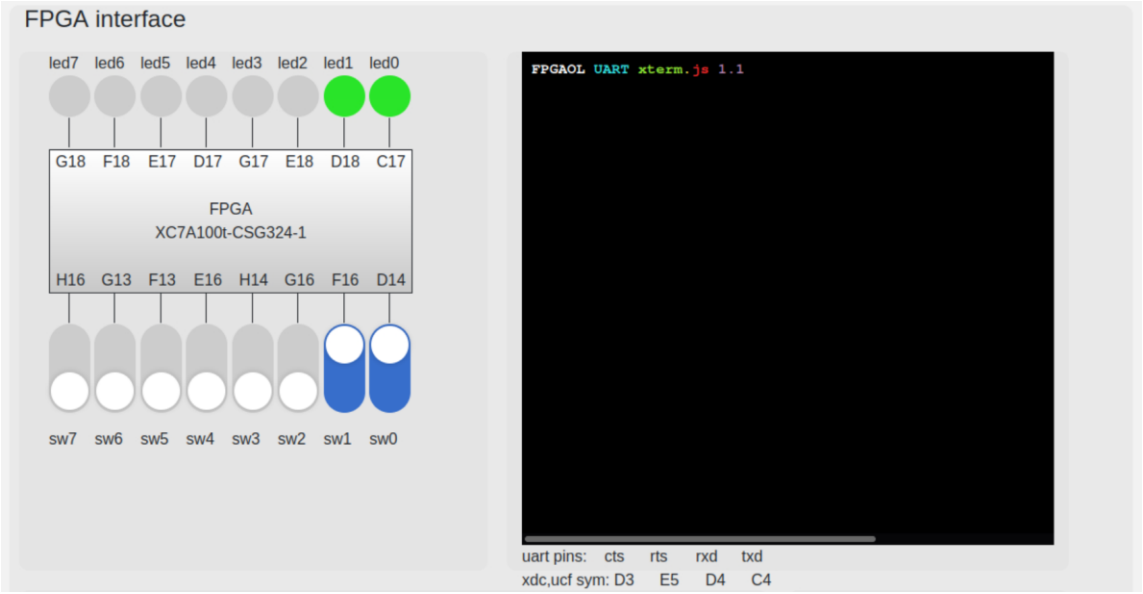


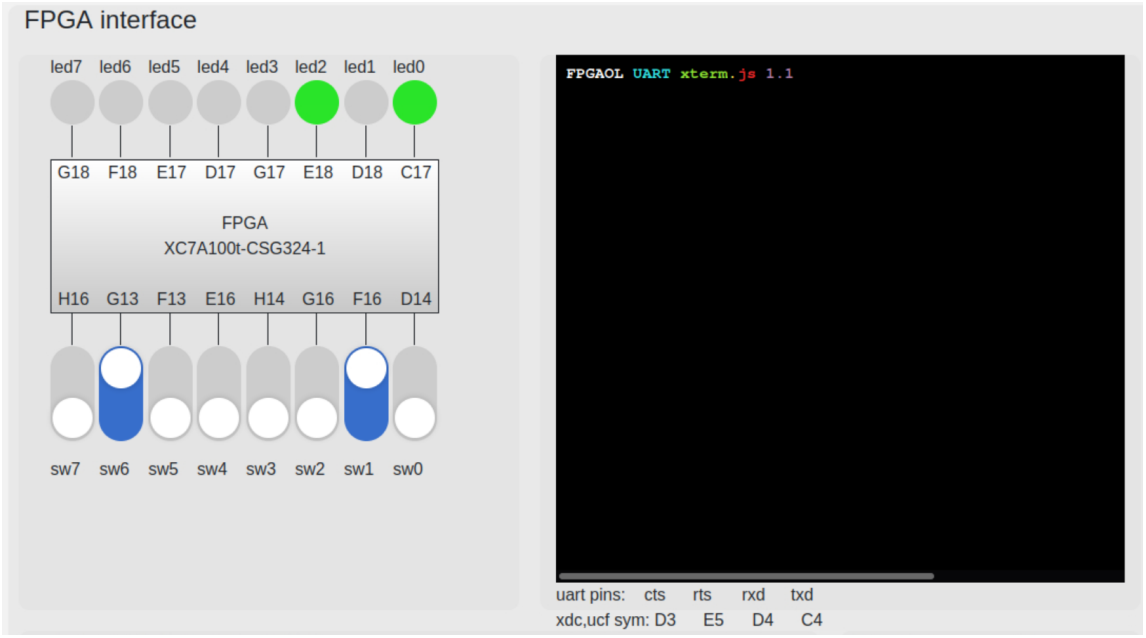
FPGA 效果图：

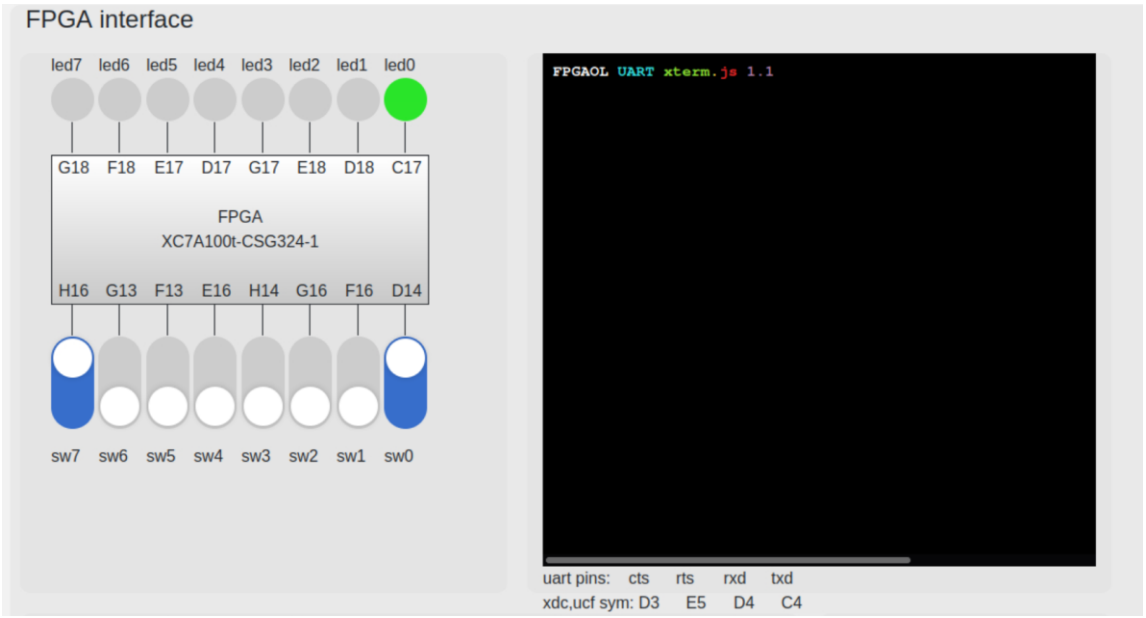
1. 6-ALU 算术逻辑运算器运算展示

以 3-2 为例：

```
assign ea = ~sel[1]& ~sel[0]  & en;//00
assign eb = ~sel[1]& sel[0]  & en;//01
assign ef = sel[1]& ~sel[0]  & en;//10
```

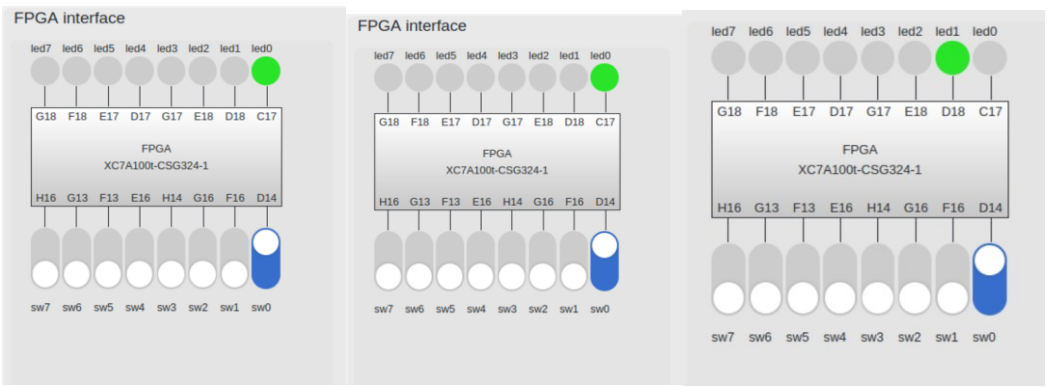


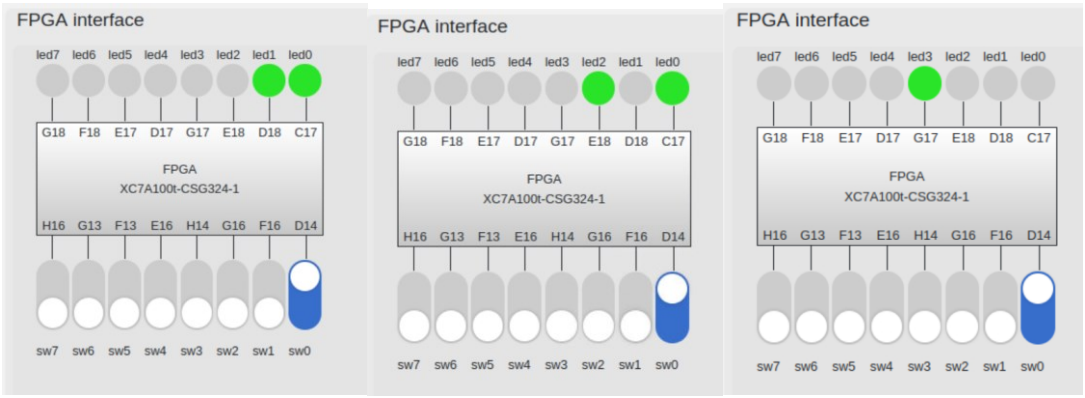




2. FLS 斐波拉契数列 bit 烧板展示

由 1 得到 8 为止：





六、心得体会：

本次实验首先以编写基本 32 位操作数 ALU 开始，对以前的知识进行复习，之后利用译码器和此 ALU 编写 6 位操作数 ALU，并练习了 Vivado 波形仿真。最后计算斐波拉契数列，在 FPGA 平台上烧写代码。

本次实验，开始写约束文件.xdc 时不知道怎么去连接以及 clk 怎么去写，后来问助教和同学才了解到这个是和我们 FPGA 管脚对应连接，属于一种硬件连接文件吧。然后就是在写代码过程中，时序逻辑可以含 reg，组合逻辑不能含 reg！！！这是在开始就要定义好的，我中途搞忘了，然后一直 bug 找了好久才找到原因……