

Lab 1 report

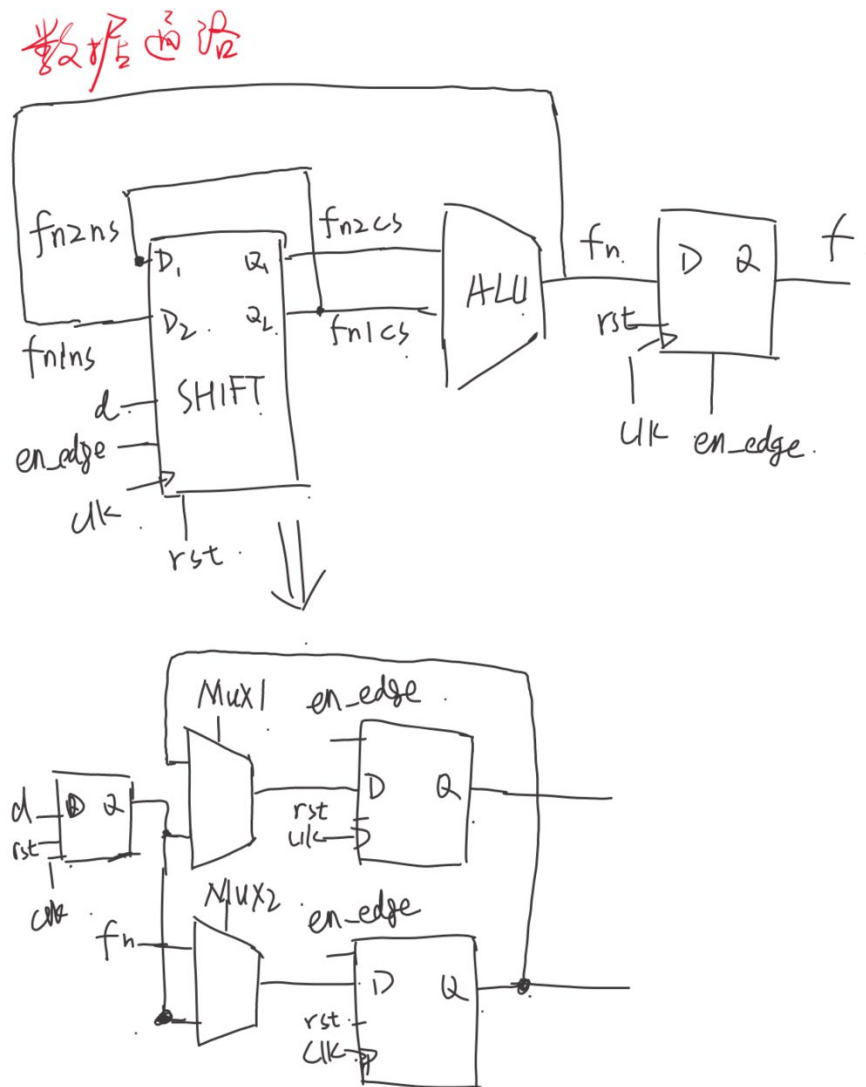
实验目的与内容

本次实验要求设计算数逻辑单元 (ALU)，正确实现其逻辑设计，并对其功能进行测试，查看分析综合后的电路图；利用设计好的 ALU 计算斐波那契数列 (FLS)，完成 FLS 的逻辑设计、功能仿真和下载测试。

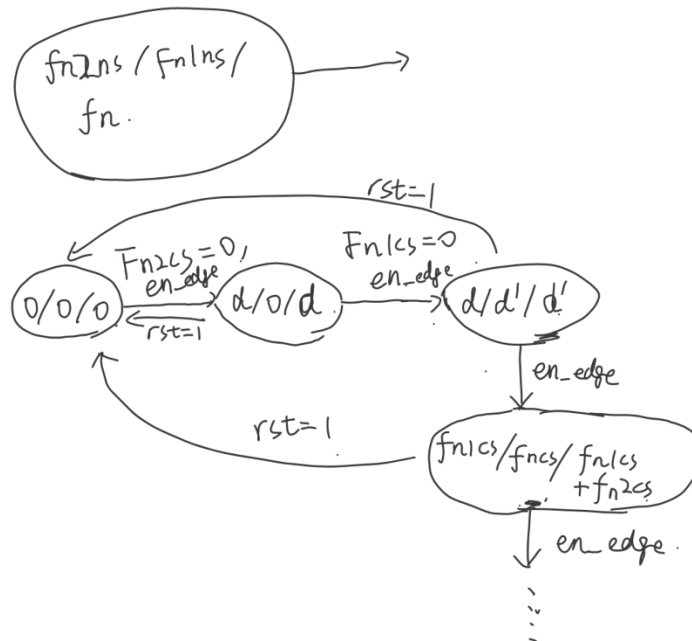
本次实验旨在帮助学生掌握算数逻辑单元的功能；掌握数据通路和有限状态机的设计方法；掌握组合电路和时序电路，以及参数化、结构化的 Verilog 描述方法。

逻辑设计

FLS 的数据通路：



FLS 状态机:



核心代码:

1、ALU 部分:

```

module ALU #(parameter WIDTH = 6)(
    input [WIDTH - 1:0] a, b, //operand
    input [3:0] func, //function
    output reg [WIDTH - 1:0] y, //the answer
    output reg of, //if overflow or not
);
//assign of = 0;
always @(*)
begin
    case(func)
        4'b0000: begin
            y = a + b;
            of = (a[WIDTH - 1] & b[WIDTH - 1] & (~y[WIDTH - 1])) | ((~a[WIDTH - 1]) & (~b[WIDTH - 1]) & y[WIDTH - 1]);
        end
        4'b0001: begin
            y = a - b;
            if(y[WIDTH - 1] == y[WIDTH - 2])
                of = 1;
            else of = 0;
        end
        4'b0010: begin
            if( a == b) y <= 1; else y <= 0;
        end
    endcase
end
  
```

```

of <= 0;end
4'b0011: begin if( a < b) y <= 1; else y <= 0; of <= 0;end
4'b0100:begin
    if($signed(a) < $signed(b)) y <= 1; else y <= 0;
    of <= 0;
end
4'b0101: begin y <= a & b; of <= 0;end
4'b0110: begin y <= a | b; of <= 0;end
4'b0111: begin y <= a ^ b; of <= 0;end
4'b1000: begin y = a >> b; of = 0;end
4'b1001: begin y = a << b; of = 0;end
default: begin y <= 0; of <= 0;end
endcase
end
endmodule

```

ALU 设计部分较为简单，选择以下几点加以讲述：

(1) 溢出 of 的求取：在加法运算中，我选择了较为复杂的分别考虑正数溢出和负数溢出的方法，而减法中，选取相对简洁的比较结果的符号位和最高位的方法。

(2) 有符号数的比较需要通过\$signed () 完成

2、FLS 部分

```

module FLS(
    input clk, rst,
    input en,
    input [6:0] d,
    output reg [6:0] f
);
reg [6:0] Fn1, Fn2, Fn;//current state
reg [6:0] fn1, fn2, fn;//next state
reg en1, en2;
wire en_edge;
wire of;

always@(posedge clk)//get edge
begin
    en1 <= en;
    en2 <= en1;
end
assign en_edge = en1 & (~en2);

always@(*)//state shifting
begin
    if(Fn2 == 0)
        begin

```

```

        fn2 = d;
        fn1 = 0;
        fn = d;
    end
    else if(Fn1 == 0)
        begin
            fn2 = Fn2;
            fn1 = d;
            fn = d;
        end
    else
        begin
            fn1 = Fn;
            fn2 = Fn1;
            fn = Fn;
        end
    end
end

```

```

ALU #(7) alu7(Fn1, Fn2, 4'b0000, Fn, of);

```

```

always@(posedge clk)//register
begin
    if(rst)
        begin
            Fn1 <= 0;
            Fn2 <= 0;
        end
    else if(en_edge)
        begin
            Fn1 <= fn1;
            Fn2 <= fn2;
        end
    else
        begin
            Fn1 <= Fn1;
            Fn2 <= Fn2;
        end
    end
end

```

```

always@(posedge clk)//output
begin
    if(rst)
        f <= 0;
    else if(en_edge)
        f <= fn;
end

```

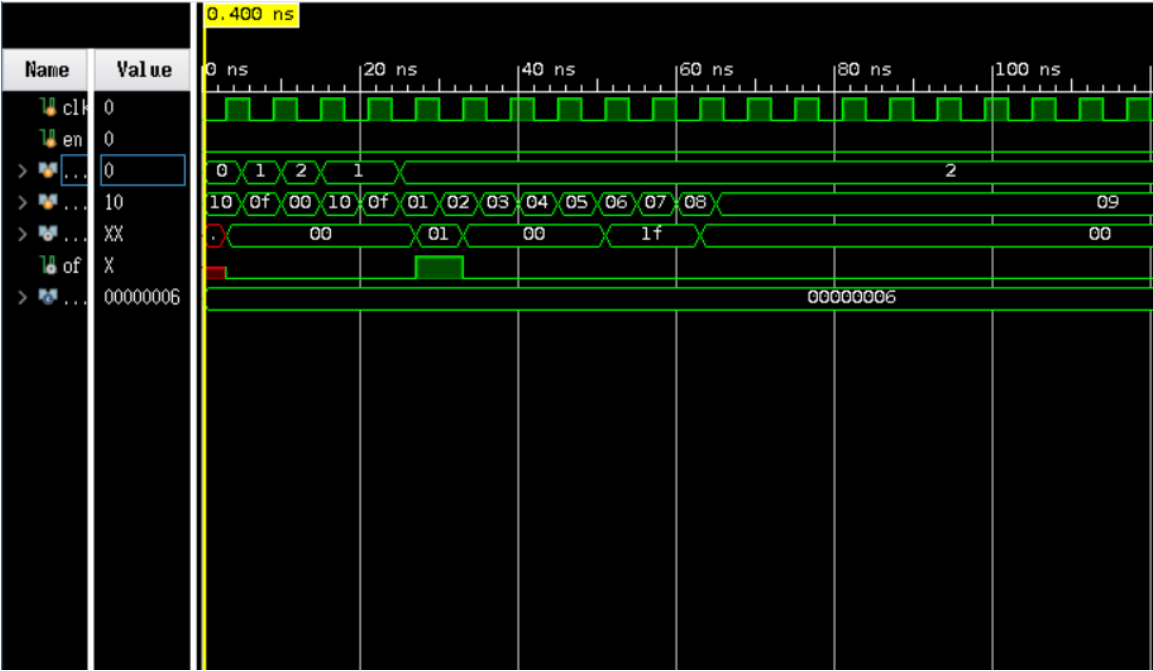
```
else
    f <= f;
end
endmodule
```

逻辑设计中取 Fn1、Fn2、Fn 作为现态，fn1、fn2、fn 作为次态。

第一部分取信号 en 的边沿，第二部分为状态的转移（组合逻辑），第三部分为寄存器，用于对现态的改变，第四部分为输出。其中 Fn 的计算调用了前面设计的 ALU 模块。

仿真结果与分析

1、ALU_test 部分仿真截图：



信号从上到下依次为：clk、en、sel、x、y、of。
仿真文件如下：

```
module alup #(parameter WIDTH = 6)(
);
    reg clk, en;
    reg [1:0] sel;
    reg [5:0] x;
    wire [5:0] y;
    wire of;

    ALU_test ALU_test(.clk(clk), .en(en), .sel(sel), .x(x), .y(y), .of(of));

    initial clk = 0;
    always #3 clk = ~clk;
```

```

initial
begin
    x = 6'b010000; sel = 2'b00; en = 0; //a
    #5 x = 6'b001111; sel = 2'b01; //b
    #5 x = 6'b000000; sel = 2'b10; //+

    #5 x = 6'b010000; sel = 2'b01; //overflow  b

    #5 x = 6'b001111; sel = 2'b01; //b
    #5 x = 6'b000001; sel = 2'b10; //-

    #5 x = 6'b000010; sel = 2'b10; //==

    #5 x = 6'b000011; sel = 2'b10; //<u
    #5 x = 6'b000100; sel = 2'b10; //<

    #5 x = 6'b000101; sel = 2'b10; //&
    #5 x = 6'b000110; sel = 2'b10; //|
    #5 x = 6'b000111; sel = 2'b10; //^

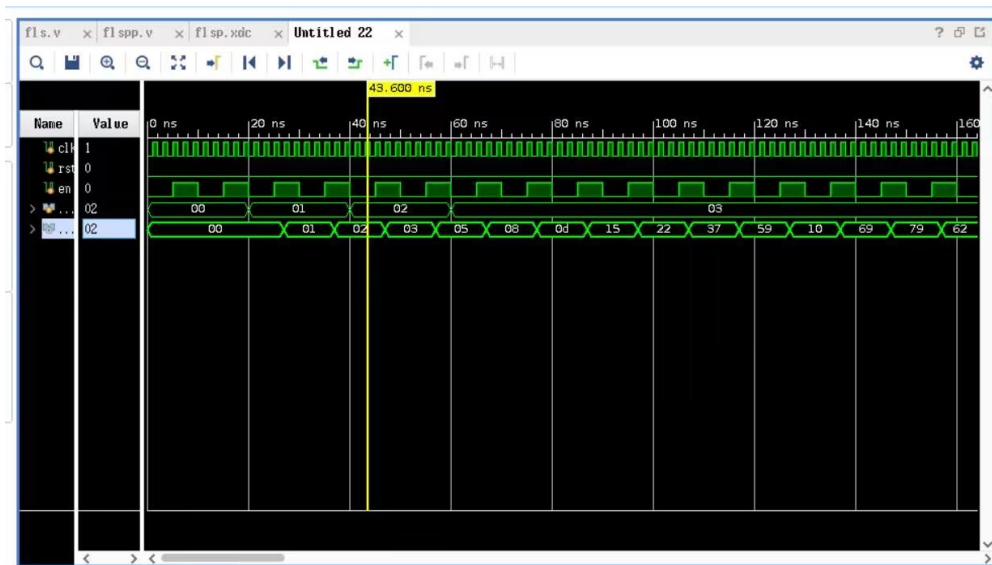
    #5 x = 6'b001000; sel = 2'b10; //>>
    #5 x = 6'b001001; sel = 2'b10; //<<

end

```

在仿真文件中测试了各个功能以及溢出情况。

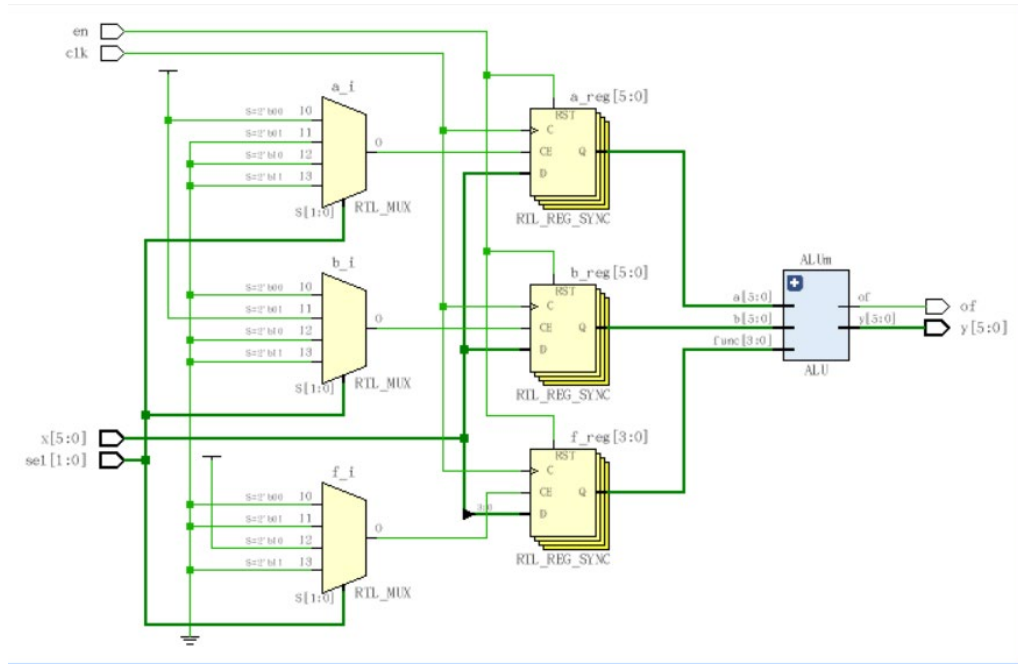
2、FLS 部分仿真



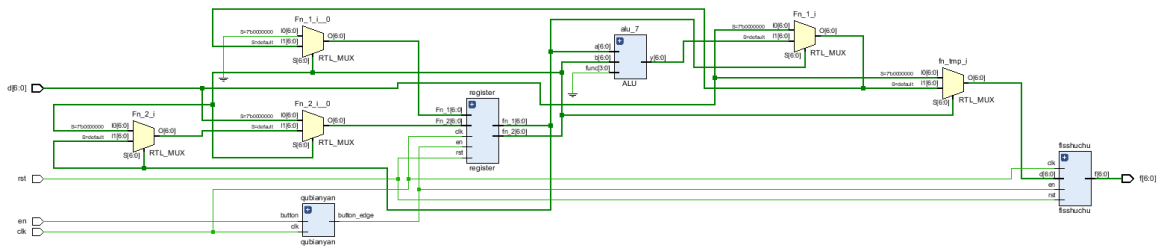
信号从上到下依次为：clk、rst、en、d、f。

电路设计与分析

ALU_test 的 RTL 电路:



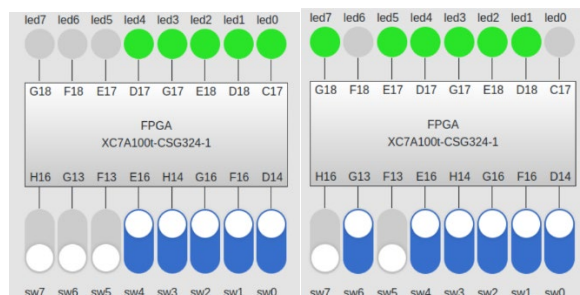
FLS 部分 RTL 电路:



测试结果与分析

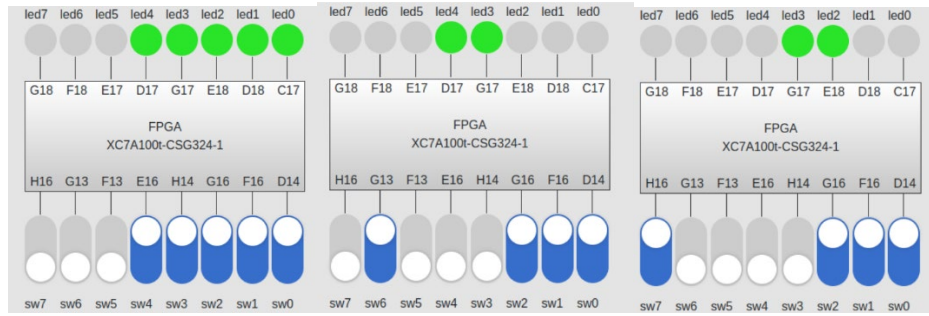
ALU_test 上板截图:

(1)



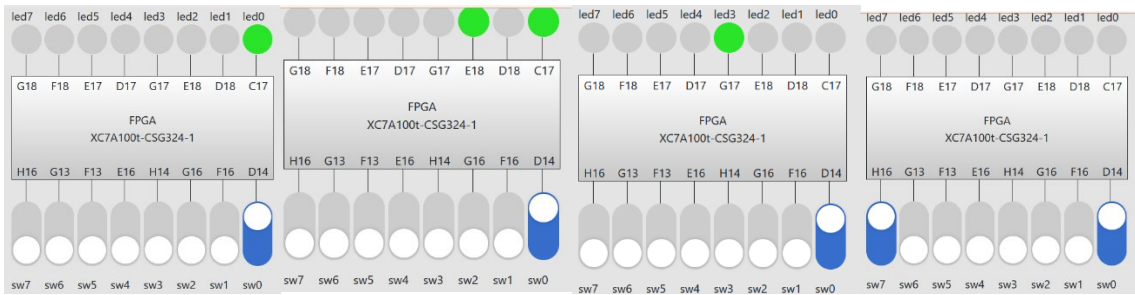
这是将 x, y 都设置为 6'b011111 时的加法运算，有溢出。

(2)



这是求取 x 、 y 的异或。

FLS 上板截图：



依次为 button 一次、五次、六次然后 reset 的截图。

总结

本次实验中，我们设计了 ALU 的 10 种逻辑功能，完成了对 ALU 加减法的溢出判断。设计了 FLS 的数据通路和状态转换图，并依次完成了 FLS 的逻辑设计（代码）。本次实验收获颇丰，不仅熟练了对 vivado 软件的使用，也加深了对数据通路和状态机的理解。