**中国科学技术大学计算机学院**

**《数字电路实验》报告**



实验题目：信号处理及有限状态机

学生姓名：\_ \_李远航\_\_ \_\_\_

学生学号： \_PB20000137\_ \_

完成日期： 2021.12.10\_ \_

计算机实验教学中心制

2020年09月

【实验题目】

**信号处理及有限状态机**

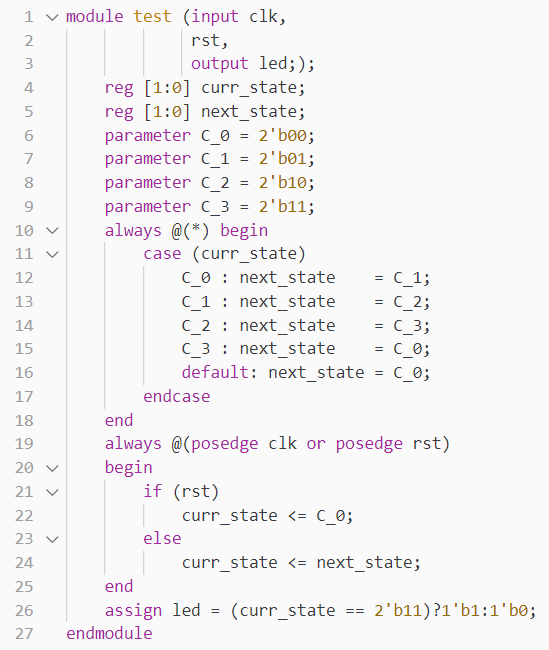
【实验目的】

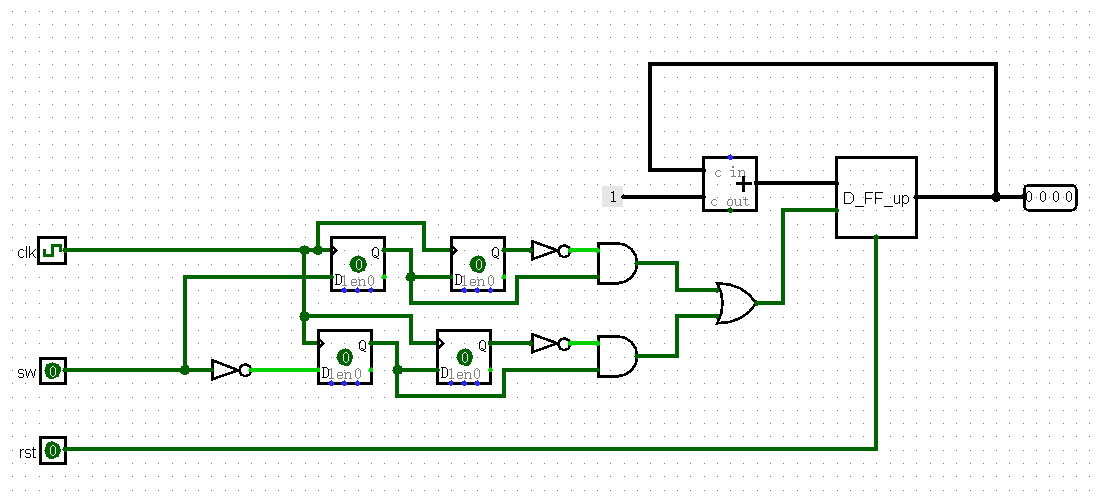
* 进一步熟悉 FPGA 开发的整体流程
* 掌握几种常见的信号处理技巧
* 掌握有限状态机的设计方法
* 能够使用有限状态机设计功能电路
* 学习时分复用的技巧，更加熟练使用verilog语言

【实验环境】

* PC一台
* Windows 10 操作系统
* Vivado 2020.02
* fpgaol.ustc.edu.cn

【实验练习】

1.  修改成三段式的verilog代码如下所示

1. 利用实验文档中的介绍，生成一个时钟周期的脉冲，同时需要检测正负，则可以再加上一个非门，电路如下所示：

1. 使用一个计数器来实现时分复用，同时额外定义reg变量来对按键信号上升沿进行捕捉，同时根据xuanze信号来进行加减计数，代码，运行截图及管口约束文件如下

    always@(posedge clk)

    begin

        flag <= count;

    end

    always@(posedge rst or posedge flag)

    begin

        if (rst)

            ans <= 8'h1f;

        else

            ans <= next;

    end

    always @(posedge clk)

    begin

        if (cnt == 4'd9)

            cnt <= 4'd0;

        else

            cnt <= cnt+4'd1;

    end

    always@(posedge clk)

        if (cnt>4'd5)

        begin

            select <= 3'b001;

            out    <= ans[7:4];

        end

        else

        begin

            select <= 3'b000;

            out    <= ans[3:0];

        end

endmodule

module fun(input clk,

           input xuanze,

           input rst,

           input count,

           output reg [3:0] out,

           output reg [2:0] select);

    reg [7:0] ans;

    reg [4:0] cnt;

    reg [7:0] next;

    reg flag;

    reg fuwei;

    initial next = 8'h1f;

    initial ans  = 8'h1f;

    initial cnt  = 4'd0;

    always@(posedge clk)

    begin

        if (xuanze)

        begin

            if (ans == 8'hff)

                next <= 8'h1f;

            else

                next <= ans+8'h1;

        end

        else

        begin

            if (ans == 8'h00)

                next <= 8'h1f;

            else

                next <= ans-8'h1;

        end

    end

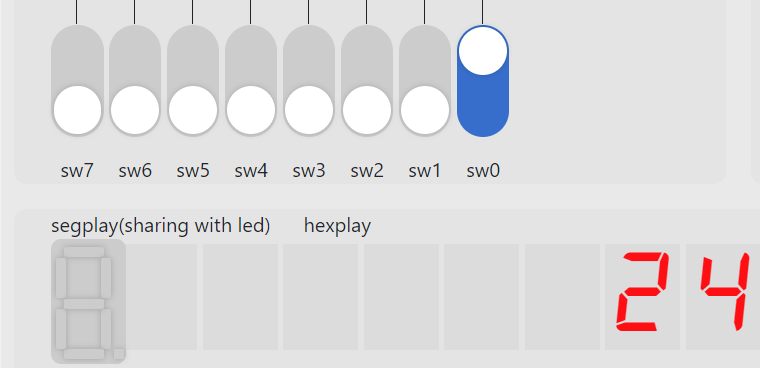


图3.1

set\_property -dict { PACKAGE\_PIN E3    IOSTANDARD LVCMOS33 } [get\_ports { clk }];

set\_property -dict { PACKAGE\_PIN D14   IOSTANDARD LVCMOS33 } [get\_ports { xuanze }];

set\_property -dict { PACKAGE\_PIN F16   IOSTANDARD LVCMOS33 } [get\_ports { rst }];

set\_property -dict { PACKAGE\_PIN A14   IOSTANDARD LVCMOS33 } [get\_ports { out[0] }];

set\_property -dict { PACKAGE\_PIN A13   IOSTANDARD LVCMOS33 } [get\_ports { out[1] }];

set\_property -dict { PACKAGE\_PIN A16   IOSTANDARD LVCMOS33 } [get\_ports { out[2] }];

set\_property -dict { PACKAGE\_PIN A15   IOSTANDARD LVCMOS33 } [get\_ports { out[3] }];

set\_property -dict { PACKAGE\_PIN B17   IOSTANDARD LVCMOS33 } [get\_ports { select[0] }];

set\_property -dict { PACKAGE\_PIN B16   IOSTANDARD LVCMOS33 } [get\_ports { select[1] }];

set\_property -dict { PACKAGE\_PIN A18   IOSTANDARD LVCMOS33 } [get\_ports { select[2] }];

set\_property -dict { PACKAGE\_PIN B18   IOSTANDARD LVCMOS33 } [get\_ports { count }];

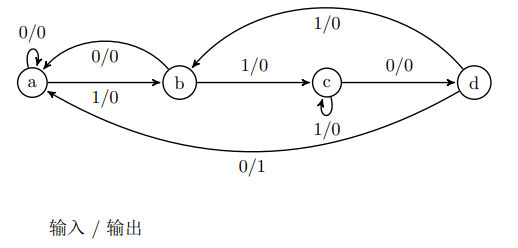
1. 通过以下状态图编写verilog代码，代码管脚约束文件及运行截图如下：

图4.1

**第一个数码管表示1100个数**

**第三个数码管表示状态：0,1,2,3对应a,b,c,d**

**最后四个数码管表示近期输入的四个数字**

**下图为输入序列：0011001110011后的情况**

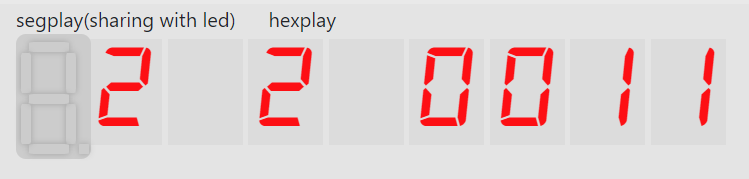


图4.2

**Verilog代码：**

   module judge(input clk,

             input string,

             input get\_num,

             output reg [2:0] sel,

             output reg [3:0] out);

    reg [4:0] cnt;

    reg [15:0] last\_num;

    initial last\_num = 16'd0;

    initial cnt <= 5'd0;

    reg [1:0] curr\_state;

    reg [1:0] next\_state;

    parameter a        = 2'b00;

    parameter b        = 2'b01;

    parameter c        = 2'b10;

    parameter d        = 2'b11;

    initial curr\_state = 2'b00;

    always@(\*)

    begin

        case(curr\_state)

            a :

            begin

                if (string == 1)

                    next\_state = b;

                else

                    next\_state = a;

            end

            b :

            begin

                if (string == 1)

                    next\_state = c;

                else

                    next\_state = a;

            end

            c :

            begin

                if (string == 1)

                    next\_state = c;

                else

                    next\_state = d;

            end

            d :

            begin

                if (string == 1)

                next\_state = b;

                else

                    next\_state = a;

            end

        endcase

    end

    reg flag;

    always@(posedge clk)

        flag <= get\_num;

    always@(posedge flag)

    begin

        last\_num[15:12] <= last\_num[11:8];

        last\_num[11:8]  <= last\_num[7:4];

        last\_num[7:4]   <= last\_num[3:0];

        last\_num[3:0]   <= {3'b0,string};

        curr\_state      <= next\_state;

    end

    reg [3:0] num\_now;

    initial num\_now = 4'b0;

    reg [3:0] state\_now;

    always@(\*)

        state\_now = {2'b00,curr\_state};

    always@(posedge flag)

    begin

        if (last\_num[11:0] == 12'h110&&string == 0)

        begin

            num\_now <= num\_now+4'b1;

        end

    end

    always@(posedge clk)

    begin

        if (cnt == 5'd24)

            cnt <= 5'd0;

        else

            cnt <= cnt+5'd1;

    end

    always@(posedge clk)

    begin

        if (cnt<5'd4)

        begin

            sel <= 3'b111;

            out <= num\_now;

        end

        else if (cnt<5'd8)

        begin

            sel <= 3'b101;

            out <= state\_now;

        end

        else if (cnt<5'd12)

        begin

            sel <= 3'b011;

            out <= last\_num[15:12];

        end

        else if (cnt<5'd16)

        begin

            sel <= 3'b010;

            out <= last\_num[11:8];

        end

        else if (cnt<5'd20)

        begin

            sel <= 3'b001;

            out <= last\_num[7:4];

        end

        else if (cnt<5'd24)

        begin

            sel <= 3'b000;

            out <= last\_num[3:0];

        end

    end

endmodule

**管脚约束文件:**

set\_property -dict { PACKAGE\_PIN E3    IOSTANDARD LVCMOS33 } [get\_ports { clk }];

set\_property -dict { PACKAGE\_PIN D14   IOSTANDARD LVCMOS33 } [get\_ports { string }];

set\_property -dict { PACKAGE\_PIN A14   IOSTANDARD LVCMOS33 } [get\_ports { out[0] }];

set\_property -dict { PACKAGE\_PIN A13   IOSTANDARD LVCMOS33 } [get\_ports { out[1] }];

set\_property -dict { PACKAGE\_PIN A16   IOSTANDARD LVCMOS33 } [get\_ports { out[2] }];

set\_property -dict { PACKAGE\_PIN A15   IOSTANDARD LVCMOS33 } [get\_ports { out[3] }];

set\_property -dict { PACKAGE\_PIN B17   IOSTANDARD LVCMOS33 } [get\_ports { sel[0] }];

set\_property -dict { PACKAGE\_PIN B16   IOSTANDARD LVCMOS33 } [get\_ports { sel[1] }];

set\_property -dict { PACKAGE\_PIN A18   IOSTANDARD LVCMOS33 } [get\_ports { sel[2] }];

set\_property -dict { PACKAGE\_PIN B18   IOSTANDARD LVCMOS33 } [get\_ports { get\_num }];

【总结与思考】

* 本次实验难度较大，任务量较多
* 进一步熟悉了FPGA的开发流程
* 学会设计并例化状态机
* 学会处理不同的信号
* 对Verilog语言有了更深的认识