

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： FPGA 实验平台及 IP 核使用

学生姓名： 李远航

学生学号： PB20000137

完成日期： 2021. 12. 02

计算机实验教学中心制

2020 年 09 月

【实验题目】

FPGA 实验平台及 IP 核使用

【实验目的】

- 熟悉 FPGAOOL 在线实验平台结构及使用
- 掌握 FPGA 开发各关键环节
- 学会使用 IP 核（知识产权核）
- 学会使用时分复用节省端口

【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. 例化一个 16*8bit 的 ROM，输入端口由四个开关控制，输出到七段数码管上，控制数码管显示与开关相对应的十六进制数字

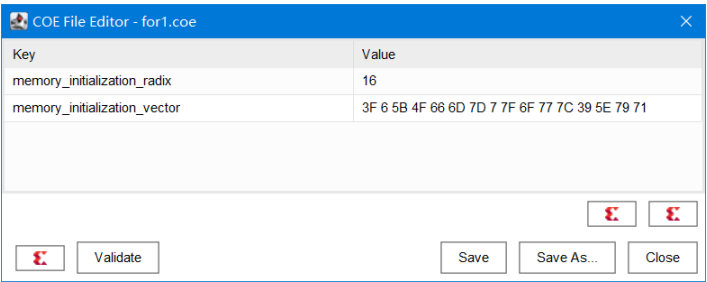


图 1.1 COE 文件

```
13 set_property -dict { PACKAGE_PIN C17 IOSTANDARD LVCMOS33 } [get_ports { spo[0] }];
14 set_property -dict { PACKAGE_PIN D18 IOSTANDARD LVCMOS33 } [get_ports { spo[1] }];
15 set_property -dict { PACKAGE_PIN E18 IOSTANDARD LVCMOS33 } [get_ports { spo[2] }];
16 set_property -dict { PACKAGE_PIN G17 IOSTANDARD LVCMOS33 } [get_ports { spo[3] }];
17 set_property -dict { PACKAGE_PIN D17 IOSTANDARD LVCMOS33 } [get_ports { spo[4] }];
18 set_property -dict { PACKAGE_PIN E17 IOSTANDARD LVCMOS33 } [get_ports { spo[5] }];
19 set_property -dict { PACKAGE_PIN F18 IOSTANDARD LVCMOS33 } [get_ports { spo[6] }];
20 set_property -dict { PACKAGE_PIN G18 IOSTANDARD LVCMOS33 } [get_ports { spo[7] }];
21
22
23 ## FPGA0L SWITCH
24
25 set_property -dict { PACKAGE_PIN D14 IOSTANDARD LVCMOS33 } [get_ports { a[0] }];
26 set_property -dict { PACKAGE_PIN F16 IOSTANDARD LVCMOS33 } [get_ports { a[1] }];
27 set_property -dict { PACKAGE_PIN G16 IOSTANDARD LVCMOS33 } [get_ports { a[2] }];
28 set_property -dict { PACKAGE_PIN H14 IOSTANDARD LVCMOS33 } [get_ports { a[3] }];
29 #set_property -dict { PACKAGE_PIN E16 IOSTANDARD LVCMOS33 } [get_ports { sw[4] }];
30 #set_property -dict { PACKAGE_PIN F13 IOSTANDARD LVCMOS33 } [get_ports { sw[5] }];
31 #set_property -dict { PACKAGE_PIN G13 IOSTANDARD LVCMOS33 } [get_ports { sw[6] }];
32 #set_property -dict { PACKAGE_PIN H16 IOSTANDARD LVCMOS33 } [get_ports { sw[7] }];
```

图 1.2 XDC 文件

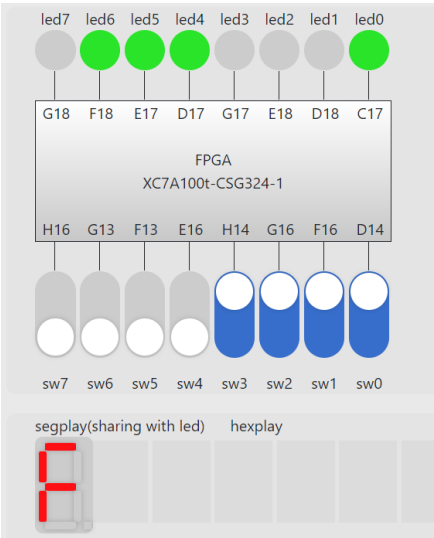


图 1.3 运行截图

2. 先写出一个模 10 的计数器，然后以 5 为分界线，分别对应第一个和第二个数码管，将开关的十六进制数值分别传给两个数码管

```

module temprunner(
    input clk,rst,
    input [7:0] sw,
    output reg [2:0] sel,
    output reg [3:0] out);
    reg [3:0] cnt;
    wire pulse;
    always@(posedge clk)
    begin
        if(rst)
            cnt<=4'h0;
        else if(cnt>=9)
            cnt <= 4'h0;
        else
            cnt <= cnt + 4'h1;
    end

    assign pulse = (cnt>4'h5);
    always@(posedge clk)
    begin
        if(pulse)
            begin
                sel<=3'b000;
                out<=sw[3:0];
            end
        else
            begin
                sel<=3'b001;
                out<=sw[7:4];
            end
    end
endmodule

```

图 2.1 Verilog 代码

```

## Clock signal
set_property -dict { PACKAGE_PIN E3      IOSTANDARD LVCMOS33 } [get_ports { clk }];

set_property -dict { PACKAGE_PIN D14     IOSTANDARD LVCMOS33 } [get_ports { sw[0] }];
set_property -dict { PACKAGE_PIN F16     IOSTANDARD LVCMOS33 } [get_ports { sw[1] }];
set_property -dict { PACKAGE_PIN G16     IOSTANDARD LVCMOS33 } [get_ports { sw[2] }];
set_property -dict { PACKAGE_PIN H14     IOSTANDARD LVCMOS33 } [get_ports { sw[3] }];
set_property -dict { PACKAGE_PIN E16     IOSTANDARD LVCMOS33 } [get_ports { sw[4] }];
set_property -dict { PACKAGE_PIN F13     IOSTANDARD LVCMOS33 } [get_ports { sw[5] }];
set_property -dict { PACKAGE_PIN G13     IOSTANDARD LVCMOS33 } [get_ports { sw[6] }];
set_property -dict { PACKAGE_PIN H16     IOSTANDARD LVCMOS33 } [get_ports { sw[7] }];

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 { rst }];

```

图 2.2 XDC 文件

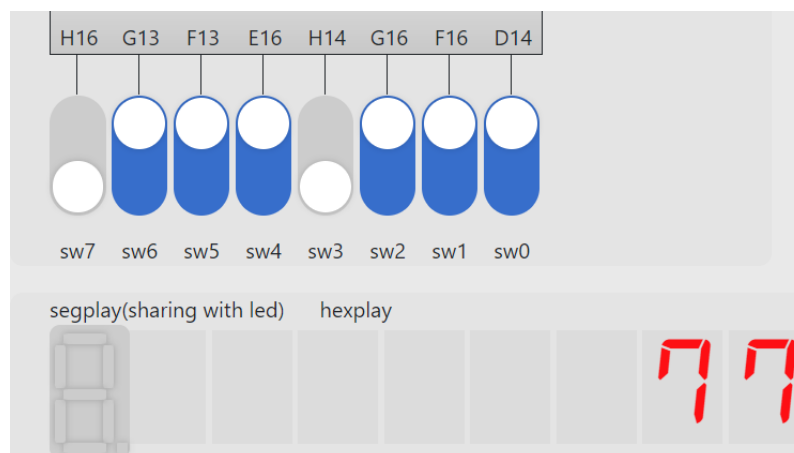


图 2.3 运行截图

- 首先利用 100MHz 的时钟信号生成 10Hz 的脉冲信号 pulse_10hz，利用该脉冲信号来记录时间，然后建立计数器来决定不同时刻输出不同信号到不同的数码管口

```

module fun(
    input clk,rst,
    output reg [3:0] out,
    output reg [2:0] select);
    reg [15:0] ti;
    reg [23:0] cnt;
    wire pulse_10hz;
    reg [4:0] for_sel;

    always@(posedge clk)
    begin
        if(for_sel>=5'd16||for_sel<5'd0)
            for_sel <= 5'd0000;
        else
            for_sel <= for_sel + 5'b0001;
    end

    always@(posedge clk)
    begin
        if(for_sel==5'd0)
            select<=3'b000;
        else if(for_sel==5'd4)
            select<=3'b001;
        else if(for_sel==5'd8)
            select<=3'b010;
        else if(for_sel==5'd12)
            select<=3'b011;
    end

    always@(posedge clk)
    begin
        if(rst)
            cnt <= 24'd0;
        else if(cnt>=24'd10000000)
            cnt <= 24'd0;
        else
            cnt <= cnt + 24'd1;
    end
    assign pulse_10hz = (cnt == 24'd0);

    always@(posedge clk)
    begin
        if(rst)
            ti<=16'h1234;
        else if(pulse_10hz)
            begin
                if(ti>=16'h9599)
                    begin
                        ti<=16'h1234;
                    end
                else if(ti[11:0]>=12'h599)
                    begin
                        ti[15:12]<=ti[15:12]+4'h1;
                        ti[11:0]<=12'h0;
                    end
                else if(ti[7:0]>=8'h99)
                    begin
                        ti[11:8]<=ti[11:8]+4'h1;
                        ti[7:0]<=8'h0;
                    end
                else if(ti[3:0]>=4'h9)
                    begin
                        ti[7:4]<=ti[7:4]+4'h1;
                        ti[3:0]<=4'h0;
                    end
                else
                    ti<=ti+16'h1;
            end
    end

    always@(posedge clk)
    begin
        if(for_sel==5'd0)
            out<=ti[3:0];
        else if(for_sel==5'd4)
            out<=ti[7:4];
        else if(for_sel==5'd8)
            out<=ti[11:8];
        else if(for_sel==5'd12)
            out<=ti[15:12];
    end
end
endmodule

```

图 3.1 Verilog 代码

```

## Clock signal
set_property -dict { PACKAGE_PIN E3      IOSTANDARD LVCMOS33 } [get_ports { clk }];

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 { rst }];

```

图 3.2 XDC 约束文件



图 3.3 运行截图

【总结与思考】

- 本次实验难度较大，任务量较多
- 学会了使用 ip 核
- 学习了时分复用的方法
- 更加熟悉了 Verilog 代码
- 希望本次实验可以增加更多的演示内容