

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：陈鸿绪

学生学号：PB21000224

完成日期：11.20

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 FPGAOL 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）

## 【实验环境】

VLAB 平台: vlab.ustc.edu.cn

FPGAOL 平台: fpgaol.ustc.edu.cn

Vivado

Logisim

## 【实验练习】

T1. 例化一个 16\*8bit 的 ROM，并对其进行初始化，输入端口由 4 个开关控制，输出端口连接到七段数码管上（七段数码管与 LED 复用相同的一组管脚），控制数码管显示与开关相对应的十六进制数字，例如四个开关输入全为零时，数码管显示“0”，输入全为 1 时，数码管显示“F”。

[解]：使用自带 IP 内核来实现 ROM，ROM 具体内容在 coe 文件中实现；Verilog 设计文件中实现 ROM 的调用；XDC 文件中实现端口对应。

Verilog 设计代码如下：

```
module test(  
input clk,rst,
```

```

input [7:0] sw,
output reg [7:0] led
    );
reg [3:0] a, dpra;
reg [7:0] d;
reg we;
wire [7:0] dpo;
always@(posedge clk)
    begin
        dpra=sw[3:0];
        we=0;
        led=dpo;
    end

dist_mem_gen_0 dist_mem_gen_0(
    .a (a),
    .d (d),
    .dpra (dpra),
    .clk (clk),
    .we (we),
    .dpo (dpo));
Endmodule

```

XDC 文件内容如下：

```

/home/ubuntu/project_12/project_12.srcs/constrs_1/new/test.xdc
1 set_property -dict { PACKAGE_PIN E3 IOSTANDARD LVCMOS33 } [get_ports { clk }];
2 set_property -dict { PACKAGE_PIN B18 IOSTANDARD LVCMOS33 } [get_ports { rst }];
3 set_property -dict { PACKAGE_PIN G18 IOSTANDARD LVCMOS33 } [get_ports { led[0] }];
4 set_property -dict { PACKAGE_PIN F18 IOSTANDARD LVCMOS33 } [get_ports { led[1] }];
5 set_property -dict { PACKAGE_PIN E17 IOSTANDARD LVCMOS33 } [get_ports { led[2] }];
6 set_property -dict { PACKAGE_PIN D17 IOSTANDARD LVCMOS33 } [get_ports { led[3] }];
7 set_property -dict { PACKAGE_PIN G17 IOSTANDARD LVCMOS33 } [get_ports { led[4] }];
8 set_property -dict { PACKAGE_PIN E18 IOSTANDARD LVCMOS33 } [get_ports { led[5] }];
9 set_property -dict { PACKAGE_PIN D18 IOSTANDARD LVCMOS33 } [get_ports { led[6] }];
10 set_property -dict { PACKAGE_PIN C17 IOSTANDARD LVCMOS33 } [get_ports { led[7] }];
11
12 set_property -dict { PACKAGE_PIN D14 IOSTANDARD LVCMOS33 } [get_ports { sw[0] }];
13 set_property -dict { PACKAGE_PIN F16 IOSTANDARD LVCMOS33 } [get_ports { sw[1] }];
14 set_property -dict { PACKAGE_PIN G16 IOSTANDARD LVCMOS33 } [get_ports { sw[2] }];
15 set_property -dict { PACKAGE_PIN H14 IOSTANDARD LVCMOS33 } [get_ports { sw[3] }];
16 set_property -dict { PACKAGE_PIN E16 IOSTANDARD LVCMOS33 } [get_ports { sw[4] }];
17 set_property -dict { PACKAGE_PIN F13 IOSTANDARD LVCMOS33 } [get_ports { sw[5] }];
18 set_property -dict { PACKAGE_PIN G13 IOSTANDARD LVCMOS33 } [get_ports { sw[6] }];
19 set_property -dict { PACKAGE_PIN H16 IOSTANDARD LVCMOS33 } [get_ports { sw[7] }];

```

Coe 文件代码如下：

```
memory_initialization_radix=16;
```

```
memory_initialization_vector=
```

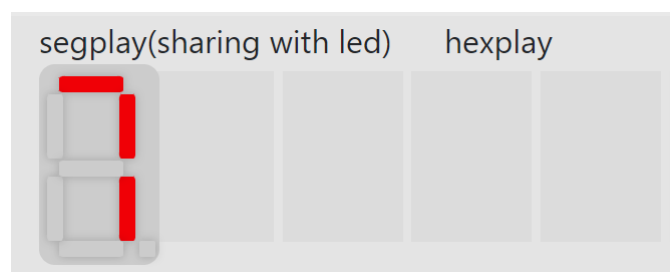
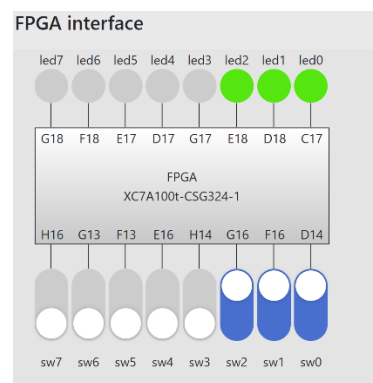
```
FC 60 DA F2
```

```
66 B6 BE E0
```

```
FE F6 EE 3E
```

```
9C 7A 9E 8E;
```

在 fpga 实验平台上烧写可以得到如下运行结果：



T2. 采用 8 个开关作为输入，两个十六进制数码管作为输出，采用时分复用的方式将开关的十六进制数值在两个数码管上显示出来，例如高四位全为 1，低四位全为 0 时，数码管显示“F0”。

[解]：四个开关控制一个十六进制数码管，另外四个开关控制另一个十六进制数码管，通过设置一个三位的 reg 变量，在 clk 时钟上沿时刻，不断增 1，若该变量最高位为 1，则通过选择器显示其中一个数码管，否则显示另一个数码管，这样的显示频率就可以使得数字可以正常时分复用显示。

Verilog 设计代码如下：

```
module test(  
    input clk,  
    input [7:0] sw,  
    output reg [3:0] temp,  
    output reg sel  
);  
reg [2:0] a=0;  
always@(posedge clk)  
begin  
    a=a+1;  
    sel=a[2];  
    if(sel==1)  
        begin
```

```

        temp<=sw[7:4];

    end

else

    begin

        temp<=sw[3:0];

    end

end

endmodule

```

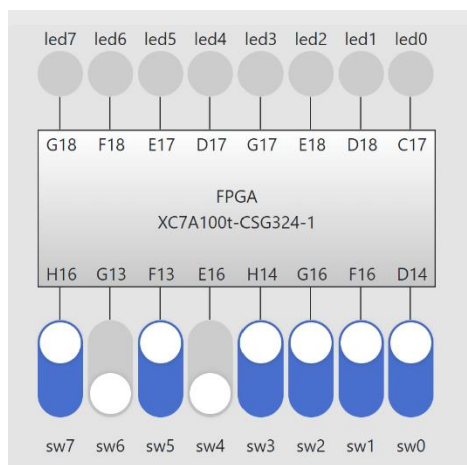
XDC 文件如下：

```

/home/ubuntu/project_13/project_13.srcs/constrs_1/new/test.xdc
1 set_property -dict { PACKAGE_PIN E3 IOSTANDARD LVCMOS33 } [get_ports { clk }];
2 set_property -dict { PACKAGE_PIN A15 IOSTANDARD LVCMOS33 } [get_ports { temp[3] }];
3 set_property -dict { PACKAGE_PIN A16 IOSTANDARD LVCMOS33 } [get_ports { temp[2] }];
4 set_property -dict { PACKAGE_PIN A13 IOSTANDARD LVCMOS33 } [get_ports { temp[1] }];
5 set_property -dict { PACKAGE_PIN A14 IOSTANDARD LVCMOS33 } [get_ports { temp[0] }];
6 |
7 set_property -dict { PACKAGE_PIN B17 IOSTANDARD LVCMOS33 } [get_ports { sel }];
8
9 set_property -dict { PACKAGE_PIN D14 IOSTANDARD LVCMOS33 } [get_ports { sw[0] }];
10 set_property -dict { PACKAGE_PIN F16 IOSTANDARD LVCMOS33 } [get_ports { sw[1] }];
11 set_property -dict { PACKAGE_PIN G16 IOSTANDARD LVCMOS33 } [get_ports { sw[2] }];
12 set_property -dict { PACKAGE_PIN H14 IOSTANDARD LVCMOS33 } [get_ports { sw[3] }];
13 set_property -dict { PACKAGE_PIN E16 IOSTANDARD LVCMOS33 } [get_ports { sw[4] }];
14 set_property -dict { PACKAGE_PIN F13 IOSTANDARD LVCMOS33 } [get_ports { sw[5] }];
15 set_property -dict { PACKAGE_PIN G13 IOSTANDARD LVCMOS33 } [get_ports { sw[6] }];
16 set_property -dict { PACKAGE_PIN H16 IOSTANDARD LVCMOS33 } [get_ports { sw[7] }];

```

在 fpga 实验平台上烧写可以得到如下运行结果：



T3. 利用本实验中的时钟管理单元或周期脉冲技术，设计一个精度为 0.1 秒的计时器，用 4 位数码管显示出来，数码管从高到低，分别表示分钟、秒钟十位、秒钟个位、十分之一秒，该计时器具有复位功能（可采用按键或开关作为复位信号），复位时计数值为 1234，即 1 分 23.4 秒

[解]：首先设计一个 0.1 秒的计时器 flag：根据时钟 clk 的频率，每当时钟上沿，s 加 1，当 s 达到一千万时，即此刻时钟过 0.1 秒，此时立马 s 恢复到 0，而 flag 在 s 恰好等于 1 时刻才为 1，其余均为 0，flag 即起到计时器的作用。再根据四位数字设计不同的计时器 t1, t2, t3, t4. 利用分时复用原理显示出四个数字.

Verilog 设计代码如下：

```
module test(  
    input clk, rst,  
    output reg [3:0] temp,  
    output reg [1:0] sel  
);  
  
reg [24:0] s=0;  
reg [3:0] t1, t2, t3, t4;  
reg [3:0] a=4'b0;  
wire flag;  
assign flag=(s==25'b01);  
always@(posedge clk)
```

```

begin
    if(s>99999998)
        s<=0;
    else
        s<=s+25' b1;
    end
always@(posedge clk or posedge rst)
begin
    if(rst) t1<=4' b0100;
    else if(flag)
        t1<=t1+4' b1;
    else if(t1>9)
        t1<=4' b0;
    end
always@(posedge clk or posedge rst)
begin
    if(rst) t2<=4' b0011;
    else if(t1>9)
        t2<=t2+4' b1;
    else if(t2>9)
        t2<=4' b0;
    end
end

```



```
always@(posedge clk or posedge rst)
```

```
begin
```

```
    if(rst) t3<=4'b0010;
```

```
    else if(t2>9)
```

```
        t3<=t3+4'b1;
```

```
    else if(t3>5)
```

```
        t3<=4'b0;
```

```
end
```

```
always@(posedge clk)
```

```
begin
```

```
    if(rst) t4<=4'b0001;
```

```
    else if(t3>5)
```

```
        t4<=t4+4'b1;
```

```
    else if(t4>9)
```

```
        t4<=4'b0;
```

```
end
```

```
always@(posedge clk)
```

```
begin
```

```
    sel=a[3:2];
```

```
    a=a+4'b1;
```

```
    if(sel==2'b00)
```

```
        temp<=t1;
```

```

        else if(sel==2'b01)

            temp<=t2;

        else if(sel==2'b10)

            temp<=t3;

        else

            temp<=t4;

    end

endmodule

```

XDC 文件如下：

```

1 set_property -dict { PACKAGE_PIN E3 IOSTANDARD LVCMOS33 } [get_ports { clk }];
2 set_property -dict { PACKAGE_PIN B18 IOSTANDARD LVCMOS33 } [get_ports { rst }];
3 set_property -dict { PACKAGE_PIN A15 IOSTANDARD LVCMOS33 } [get_ports { temp[3] }];
4 set_property -dict { PACKAGE_PIN A16 IOSTANDARD LVCMOS33 } [get_ports { temp[2] }];
5 set_property -dict { PACKAGE_PIN A13 IOSTANDARD LVCMOS33 } [get_ports { temp[1] }];
6 set_property -dict { PACKAGE_PIN A14 IOSTANDARD LVCMOS33 } [get_ports { temp[0] }];
7
8 set_property -dict { PACKAGE_PIN B17 IOSTANDARD LVCMOS33 } [get_ports { sel[0] }];
9 set_property -dict { PACKAGE_PIN B16 IOSTANDARD LVCMOS33 } [get_ports { sel[1] }];

```

在 fpga 实验平台上烧写可以得到如下运行结果：

运行过程中随机截图：



Button 按下后截图：



## 【总结与思考】

实验难度: 较难

实验任务: 适中

实验收获: 巨大