

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 原理及 Vi vado 综合

学生姓名：Ouedraogo Ezekiel B.

学生学号：PL19215001

完成日期：2020/12/01

计算机实验教学中心制

2020 年 10 月

【实验题目】

FPGA 原理及 Vi vado 综合

【实验目的】

了解 FPGA 工作原理

了解 Verilog 文件和约束文件在 FPGA 开发中的作用

学会使用 Vi vado 进行 FPGA 开发的完整流程

【实验环境】

VLAB 平台: vlab.ustc.edu.cn

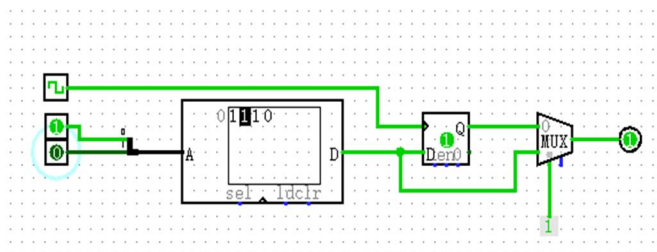
FPGA0L 实验平台: fpgaol.ustc.edu.cn

Logi si m

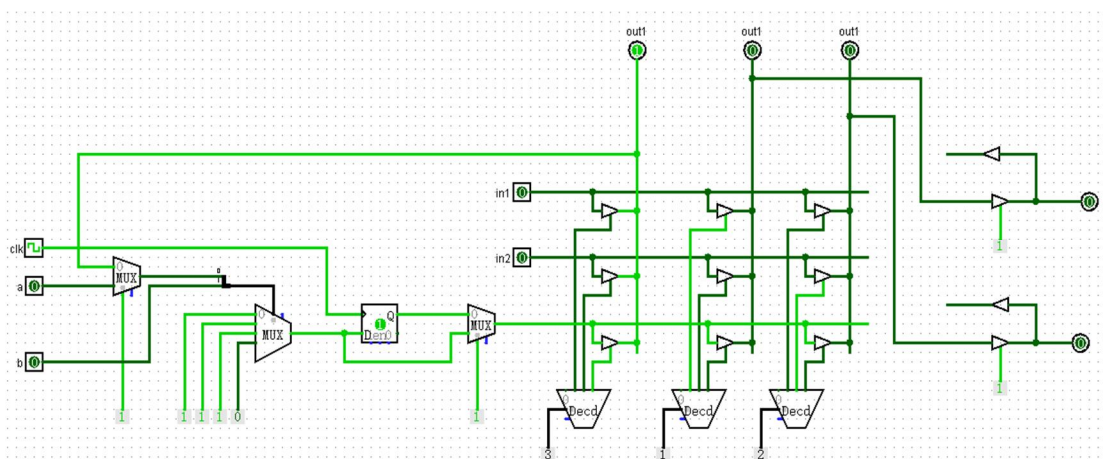
Vi vado 工具

【实验过程】

1. 可编程逻辑单元



2. 交叉互连矩阵



3. Vivado 综合

Verilog 设计代码

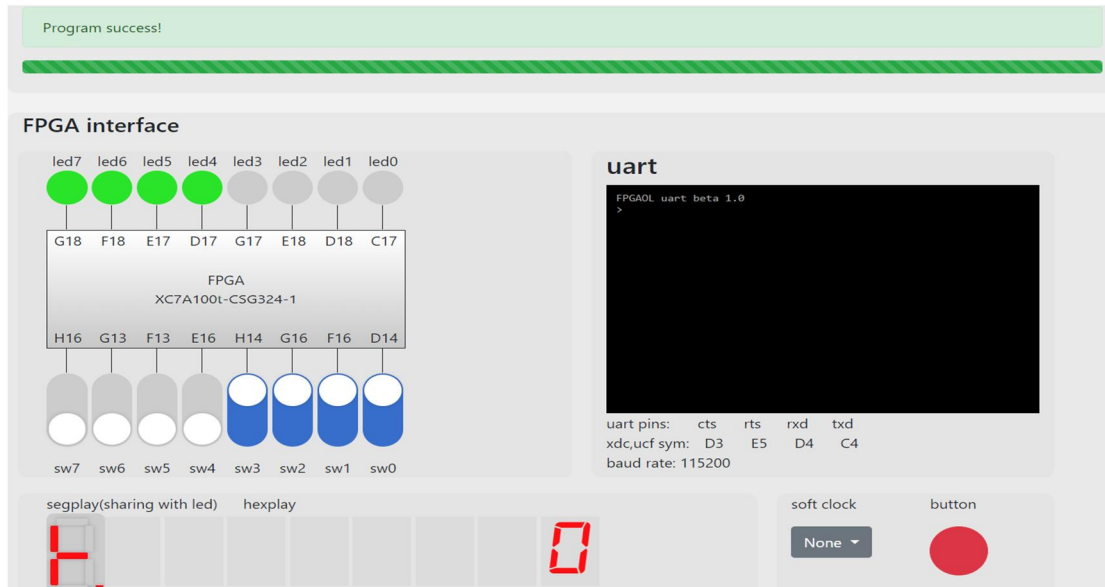
```
module test(  
    input clk,  
    input rst,  
    input [7:0] sw,  
    output reg [7:0] led  
);  
always@(posedge clk or posedge rst)  
begin  
    if(rst) led = 8'b0;  
    else led = {sw[0],sw[1],sw[2],sw[3],sw[4],sw[5],sw[6],sw[7]};  
end  
endmodule
```

管脚约束文件中的信息。

```
1 ## Clock signal  
2 set_property -dict { PACKAGE_PIN E3 IOSTANDARD LVCMOS33 } [get_ports { clk }]; #IO_L12P_T1_MRCC_35 Sch=clk100mhz  
3 #create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports {CLK100MHZ}];  
4 ## FPGAOLED BUTTON & SOFT_CLOCK  
5 set_property -dict { PACKAGE_PIN B18 IOSTANDARD LVCMOS33 } [get_ports { rst }];  
6 ## FPGAOLED LED (single-digit-SEGPLAY)  
7 set_property -dict { PACKAGE_PIN C17 IOSTANDARD LVCMOS33 } [get_ports { led[0] }];  
8 set_property -dict { PACKAGE_PIN D18 IOSTANDARD LVCMOS33 } [get_ports { led[1] }];  
9 set_property -dict { PACKAGE_PIN E18 IOSTANDARD LVCMOS33 } [get_ports { led[2] }];  
10 set_property -dict { PACKAGE_PIN G17 IOSTANDARD LVCMOS33 } [get_ports { led[3] }];  
11 set_property -dict { PACKAGE_PIN D17 IOSTANDARD LVCMOS33 } [get_ports { led[4] }];  
12 set_property -dict { PACKAGE_PIN E17 IOSTANDARD LVCMOS33 } [get_ports { led[5] }];  
13 set_property -dict { PACKAGE_PIN F18 IOSTANDARD LVCMOS33 } [get_ports { led[6] }];  
14 set_property -dict { PACKAGE_PIN G18 IOSTANDARD LVCMOS33 } [get_ports { led[7] }];  
15  
16 ## FPGAOLED SWITCH  
17 set_property -dict { PACKAGE_PIN D14 IOSTANDARD LVCMOS33 } [get_ports { sw[0] }];  
18 set_property -dict { PACKAGE_PIN F16 IOSTANDARD LVCMOS33 } [get_ports { sw[1] }];  
19 set_property -dict { PACKAGE_PIN G16 IOSTANDARD LVCMOS33 } [get_ports { sw[2] }];  
20 set_property -dict { PACKAGE_PIN H14 IOSTANDARD LVCMOS33 } [get_ports { sw[3] }];  
21 set_property -dict { PACKAGE_PIN E16 IOSTANDARD LVCMOS33 } [get_ports { sw[4] }];  
22 set_property -dict { PACKAGE_PIN F13 IOSTANDARD LVCMOS33 } [get_ports { sw[5] }];  
23 set_property -dict { PACKAGE_PIN G13 IOSTANDARD LVCMOS33 } [get_ports { sw[6] }];  
24 set_property -dict { PACKAGE_PIN H16 IOSTANDARD LVCMOS33 } [get_ports { sw[7] }];
```

4. 烧写 FPGA

结果如下



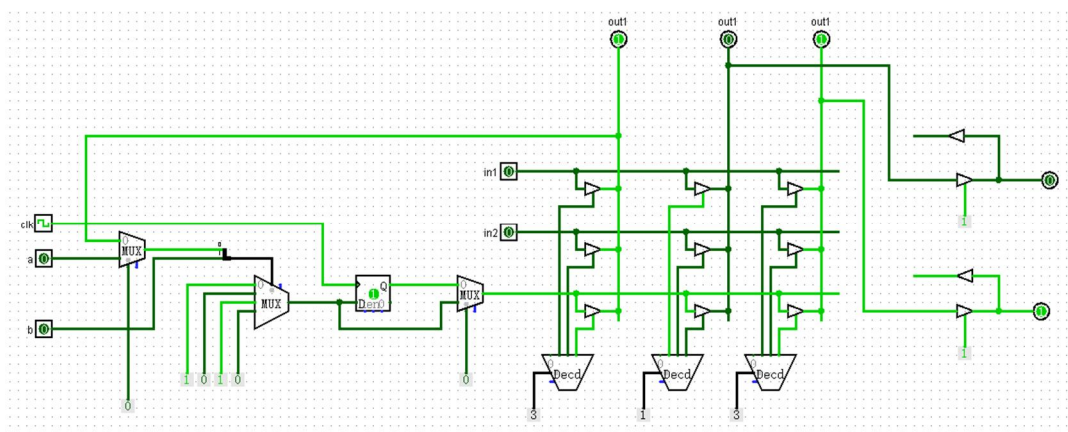
【实验练习】

1. 实现代码，并将其输出到引脚 B 上

代码

```
module test(input clk,output reg a);
always@(posedge clk)
    a <= a ^ 1'b1;
endmodule
```

电路



2. 修改实验中 给出的 XDC 文件，使开关和 LED 一一对应

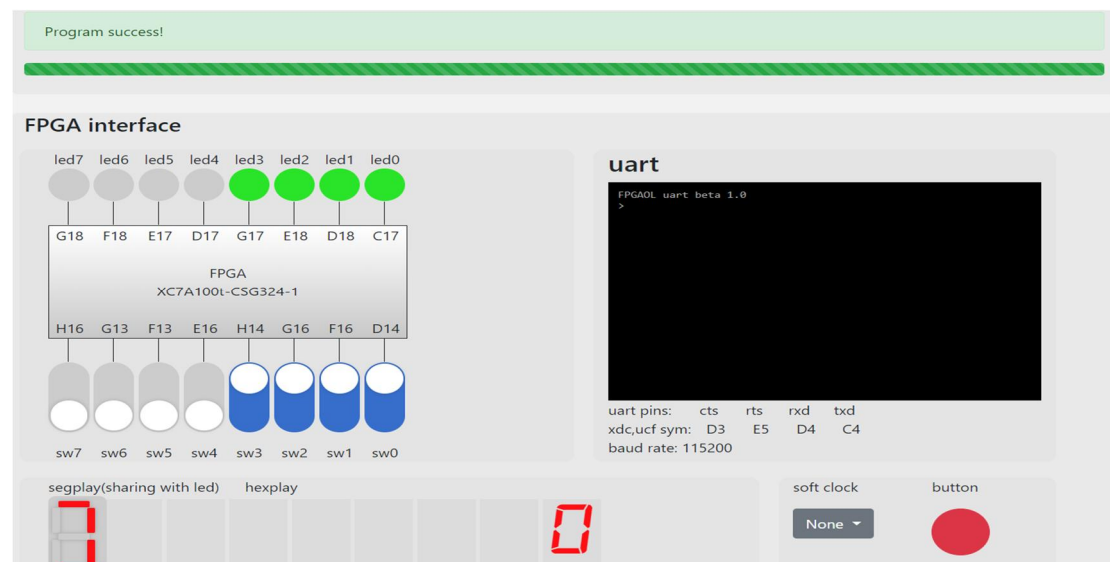
XDC 文件

```

1  ## Clock signal
2  set_property -dict { PACKAGE_PIN E3   IOSTANDARD LVCMOS33 } [get_ports { clk }]; #IO_L12P_T1_MRCC_35 Sch=clk100mhz
3  #create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports {CLK100MHZ}];
4  ## FPGAOL BUTTON & SOFT_CLOCK
5  set_property -dict { PACKAGE_PIN B18   IOSTANDARD LVCMOS33 } [get_ports { rst }];
6  ## FPGAOL LED (single-digit-SEGPLAY)
7  set_property -dict { PACKAGE_PIN C17   IOSTANDARD LVCMOS33 } [get_ports { led[0] }];
8  set_property -dict { PACKAGE_PIN D18   IOSTANDARD LVCMOS33 } [get_ports { led[1] }];
9  set_property -dict { PACKAGE_PIN E18   IOSTANDARD LVCMOS33 } [get_ports { led[2] }];
10 set_property -dict { PACKAGE_PIN G17   IOSTANDARD LVCMOS33 } [get_ports { led[3] }];
11 set_property -dict { PACKAGE_PIN D17   IOSTANDARD LVCMOS33 } [get_ports { led[4] }];
12 set_property -dict { PACKAGE_PIN E17   IOSTANDARD LVCMOS33 } [get_ports { led[5] }];
13 set_property -dict { PACKAGE_PIN F18   IOSTANDARD LVCMOS33 } [get_ports { led[6] }];
14 set_property -dict { PACKAGE_PIN G18   IOSTANDARD LVCMOS33 } [get_ports { led[7] }];
15
16 ## FPGAOL SWITCH
17 set_property -dict { PACKAGE_PIN D14   IOSTANDARD LVCMOS33 } [get_ports { sw[0] }];
18 set_property -dict { PACKAGE_PIN F16   IOSTANDARD LVCMOS33 } [get_ports { sw[1] }];
19 set_property -dict { PACKAGE_PIN G16   IOSTANDARD LVCMOS33 } [get_ports { sw[2] }];
20 set_property -dict { PACKAGE_PIN H14   IOSTANDARD LVCMOS33 } [get_ports { sw[3] }];
21 set_property -dict { PACKAGE_PIN E16   IOSTANDARD LVCMOS33 } [get_ports { sw[4] }];
22 set_property -dict { PACKAGE_PIN F13   IOSTANDARD LVCMOS33 } [get_ports { sw[5] }];
23 set_property -dict { PACKAGE_PIN G13   IOSTANDARD LVCMOS33 } [get_ports { sw[6] }];
24 set_property -dict { PACKAGE_PIN H16   IOSTANDARD LVCMOS33 } [get_ports { sw[7] }];

```

结果



3. 设计一个 30 位计数器

代码

```

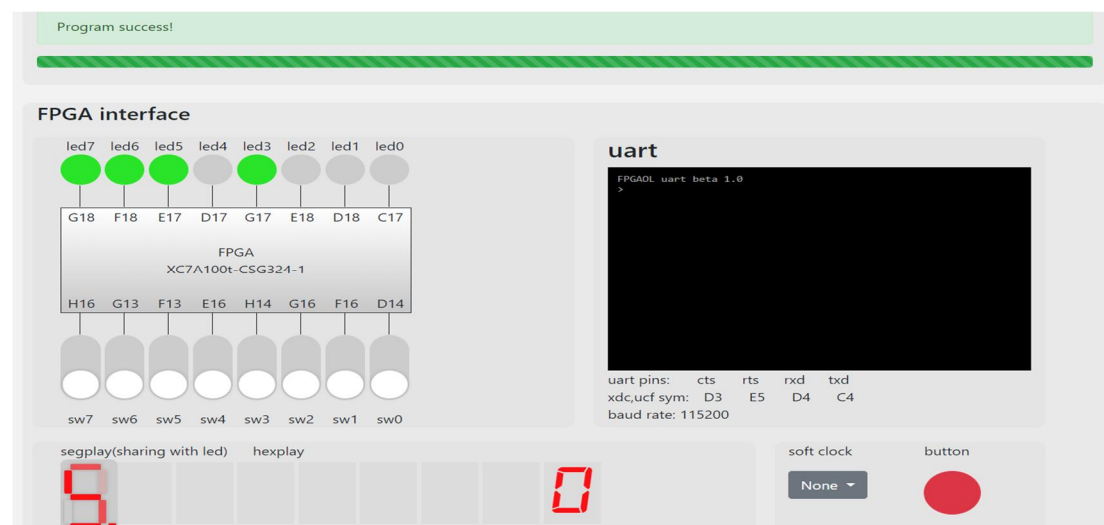
module count30(
    input clk,
    output reg [7:0] out
);
    reg [29:0] tmp;
    always@(posedge clk)
    begin
        tmp = tmp + 30'b1;
        out = tmp[29:22];
    end
endmodule

```


XDC 文件

```
1 ## Clock signal
2 set_property -dict { PACKAGE_PIN E3 IOSTANDARD LVCMOS33 } [get_ports { clk }]; #IO_L12P_T1_MRCC_35 Sch=clk100mhz
3 #create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports {CLK100MHZ}];
4 ## FPGAOL LED (single-digit-SEGPLAY)
5 set_property -dict { PACKAGE_PIN C17 IOSTANDARD LVCMOS33 } [get_ports { out[0] }];
6 set_property -dict { PACKAGE_PIN D18 IOSTANDARD LVCMOS33 } [get_ports { out[1] }];
7 set_property -dict { PACKAGE_PIN E18 IOSTANDARD LVCMOS33 } [get_ports { out[2] }];
8 set_property -dict { PACKAGE_PIN G17 IOSTANDARD LVCMOS33 } [get_ports { out[3] }];
9 set_property -dict { PACKAGE_PIN D17 IOSTANDARD LVCMOS33 } [get_ports { out[4] }];
10 set_property -dict { PACKAGE_PIN E17 IOSTANDARD LVCMOS33 } [get_ports { out[5] }];
11 set_property -dict { PACKAGE_PIN F18 IOSTANDARD LVCMOS33 } [get_ports { out[6] }];
12 set_property -dict { PACKAGE_PIN G18 IOSTANDARD LVCMOS33 } [get_ports { out[7] }];
```

结果



将该计数器改成 32 位的代码如下

```
module count(
    input clk,
    output reg [7:0] out
);
    reg [31:0] tmp;
    always@(posedge clk)
    begin
        tmp = tmp + 32'b1;
        out = tmp[31:24];
    end
endmodule
```

与前面的运行结果进行对比 32 位计数器数得慢一些。是因为 32 位的最大值比 30 位的最大值还大。

【总结与思考】

通过本次实验咱们了解了 FPGA 工作原理, Verilog 文件和约束文件在 FPGA 开发中的作用, 也学会了如何使用 Vivado 进行 FPGA 开发的完整流程。