

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 Vivado 综合

学生姓名：谭骏飞

学生学号：PB20061276

完成时间：2021.11.24

实验题目

FPGA 原理及 Vivado 综合

实验环境

Vivado 2020.2

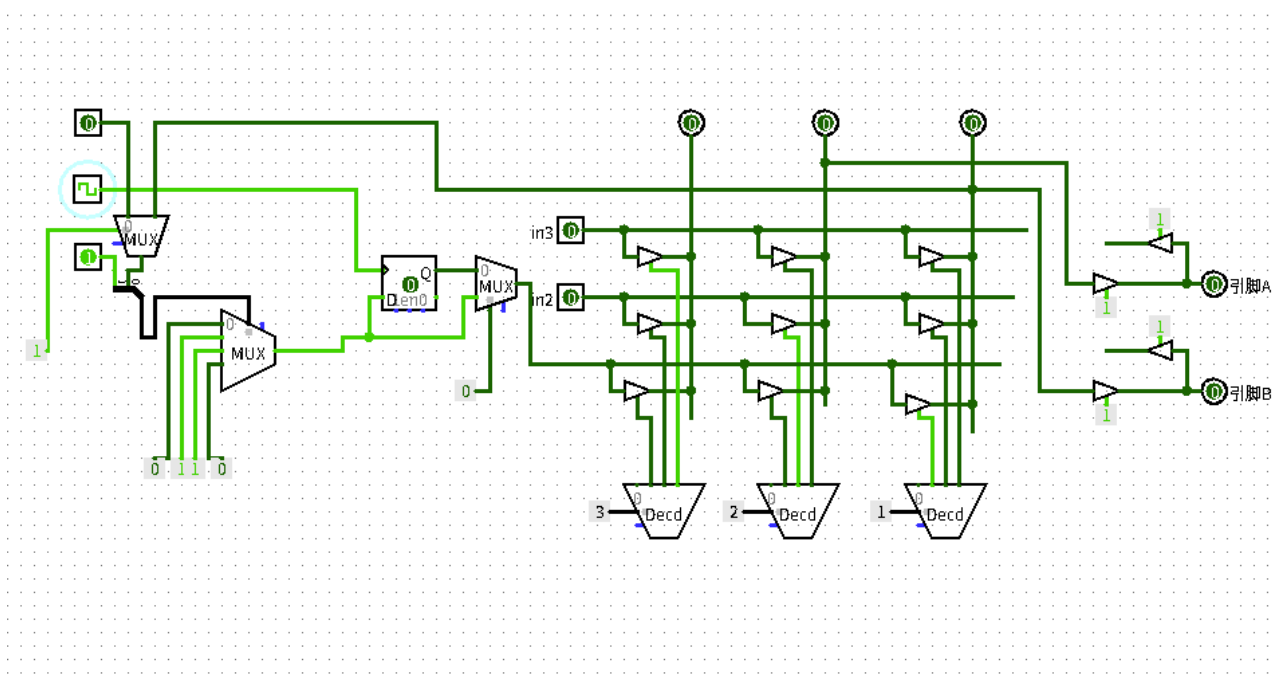
FPGAOL实验平台

实验练习

题目 **1**. 请通过实验中给出的可编程逻辑单元、交叉互连矩阵及 **IOB** 电路图，实现如下代码，并将其输出到引脚 **B** 上。给出配置数据和电路截图。

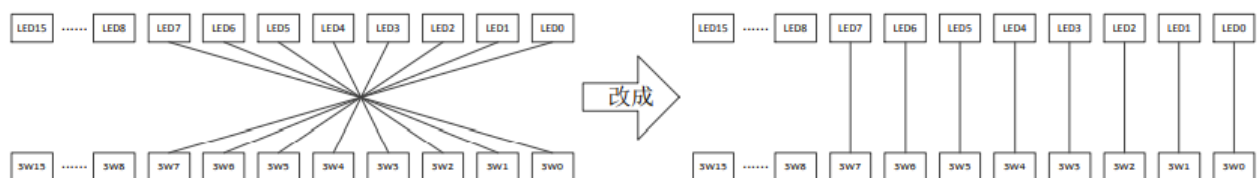
```
module test(input clk,output reg a);  
  always@(posedge clk)  
    a <= a ^ 1'b1;  
endmodule
```

由题意画出的电路图如下：



分析：代码给出了一个触发器，触发器满足每遇到一个时钟上升沿，触发器的输出都取反一次，即与1进行异或。因此在输入端的2选1数据选择器的sel信号置为1，表明选择上一次的输出信号作为新的输入信号。同时LUT供选择的信号根据00、01、10、11的异或结果分别设为0、1、1、0，以产生异或门的效果，并且将接着LUT输出端口选择器的选择信号置0，是的输出信号得以被寄存。此外，在交叉互连矩阵中，将最右侧输出信号线（亦即接着引脚B的信号线）与带有触发器输出信号的信号线之间的三态门设为有效，实现将触发器的输出信号导到引脚B的目的。

题目 2. 实验中的开关和 **LED** 的对应关系是相反的，即最左侧的开关 控制最右侧的 **LED**，最右侧的开关控制最左侧的 **LED**，请修改实验中 给出的 **XDC** 文件，使开关和 **LED** 一一对应（最左侧的开关控制最左侧 的 **LED**），如下图所示。



首先构建如下的**Verilog**设计文件，其中一个**sw**对应一个**led**的亮灭。

```
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'haa;
    else
        led <= sw;
end

endmodule

```

再将约束文件的内容改为如下的样子（将每个**led**、**sw**的管脚与相应的**led**、**sw**端口对应）。

```

## This file is a general .xdc for FPGAOL_BOARD (adopted from
Nexys4 DDR Rev. C)
## To use it in a project:
## - uncomment the lines corresponding to used pins
## - rename the used ports (in each line, after get_ports)
according to the top level signal names in the project

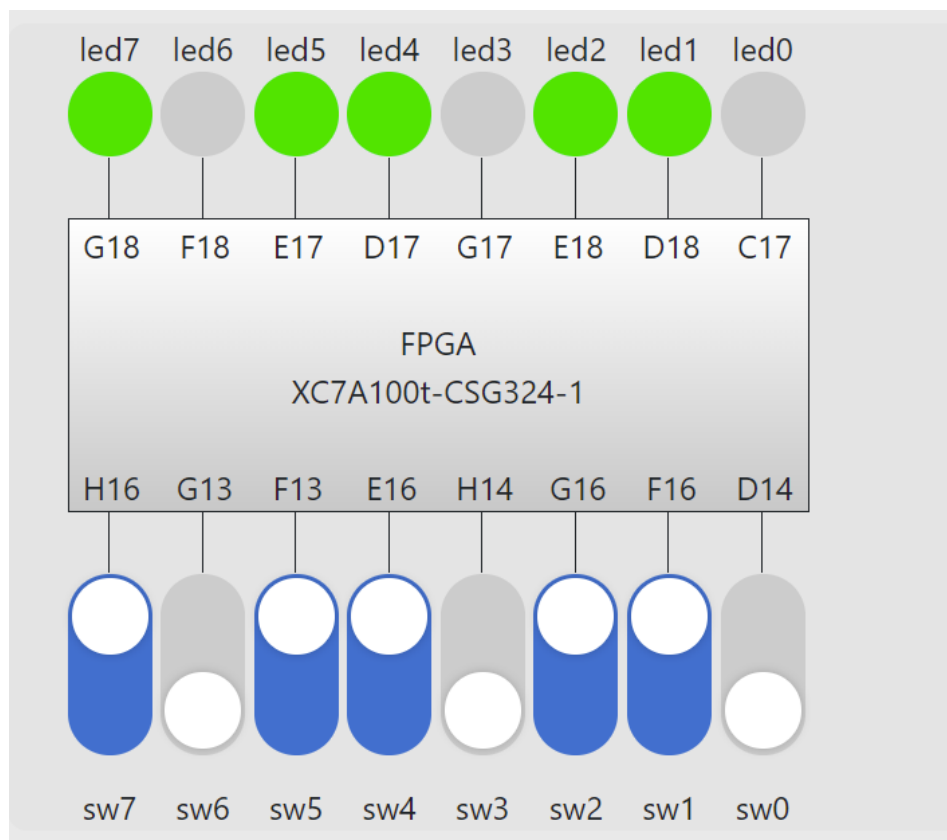
## Clock signal
set_property -dict { PACKAGE_PIN E3      IOSTANDARD LVCMOS33 }
[get_ports { clk }]; #IO_L12P_T1_MRCC_35 Sch=clk100mhz
#create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5}
[get_ports { CLK100MHZ }];
## FPGAOL BUTTON & SOFT_CLOCK
set_property -dict { PACKAGE_PIN B18      IOSTANDARD LVCMOS33 }
[get_ports { rst }];
## FPGAOL LED (single-digit-SEGPLAY)
set_property -dict { PACKAGE_PIN C17      IOSTANDARD LVCMOS33 }
[get_ports { led[0] }];
set_property -dict { PACKAGE_PIN D18      IOSTANDARD LVCMOS33 }
[get_ports { led[1] }];
set_property -dict { PACKAGE_PIN E18      IOSTANDARD LVCMOS33 }
[get_ports { led[2] }];
set_property -dict { PACKAGE_PIN G17      IOSTANDARD LVCMOS33 }
[get_ports { led[3] }];
set_property -dict { PACKAGE_PIN D17      IOSTANDARD LVCMOS33 }
[get_ports { led[4] }];
set_property -dict { PACKAGE_PIN E17      IOSTANDARD LVCMOS33 }
[get_ports { led[5] }];

```

```
set_property -dict { PACKAGE_PIN F18    IOSTANDARD LVCMOS33 }  
[get_ports { led[6] }];  
set_property -dict { PACKAGE_PIN G18    IOSTANDARD LVCMOS33 }  
[get_ports { led[7] }];  
  
## FPGAOL SWITCH  
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] }];
```

将生成的**test.bit**文件放到**FPGAOL**实验平台网站上进行烧写后的运行结果如下。

（注：生成时要确保文件所在目录的路径是全英文的）



题目 3. 设计一个 **30** 位计数器，每个时钟周期加 **1**，用右侧的 **8** 个 **LED** 表示计数器的高 **8** 位，观察实际运行结果。将该计数器改成 **32** 位，将高 **8** 位输出到 **LED**，与前面的运行结果进行对比，分析结果及时钟信号在其中所起的作用。

3.1 对于30位计数器

在Verilog设计文件中写入以下内容。

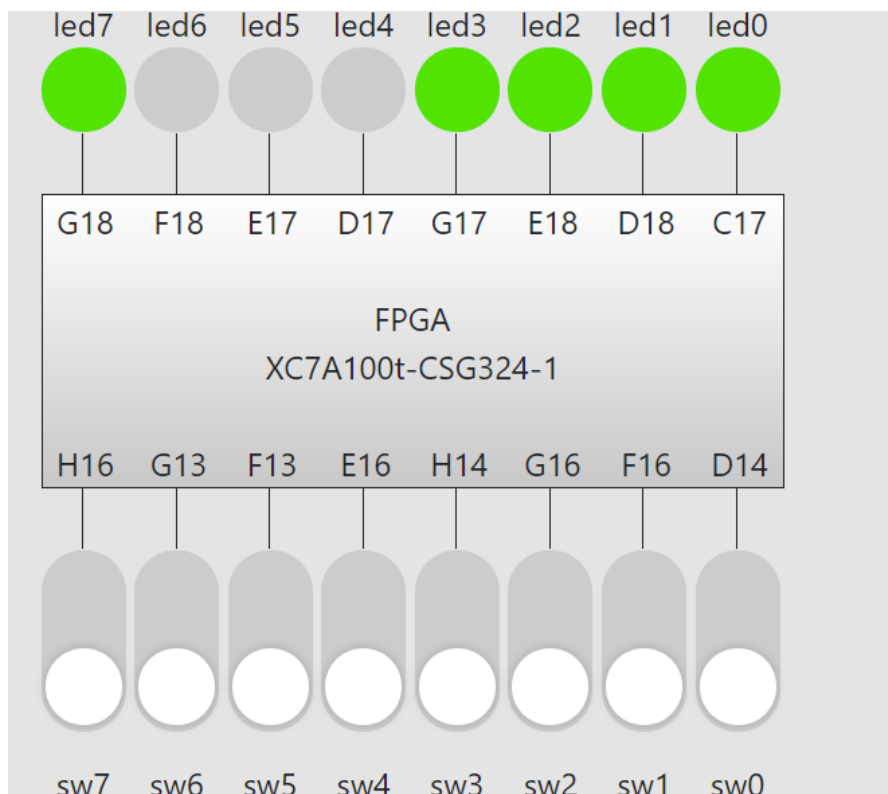
```
module test(
    input clk,
    input rst,
    output [7:0] counter
);
    reg [29:0] temp;
    assign counter = temp[29:22]; //将temp的高八位赋给counter
    always @(posedge clk or posedge rst) begin
        if(rst)
            temp<=1'b0;
        else
            temp<=temp+1'b1; //temp达到全1后再加1会自动回到全0状态
    end
endmodule
```

实验要求显示高八位，因此在约束文件中将**counter**的[0:7]与**led0~7**相对应，下面给出主体部分。

```
## This file is a general .xdc for FPGAOL_BOARD (adopted from
Nexys4 DDR Rev. C)
## To use it in a project:
## - uncomment the lines corresponding to used pins
## - rename the used ports (in each line, after get_ports)
according to the top level signal names in the project

## Clock signal
set_property -dict { PACKAGE_PIN E3      IOSTANDARD LVCMOS33 }
[get_ports { clk }]; #IO_L12P_T1_MRCC_35 Sch=clk100mhz
#create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5}
[get_ports { CLK100MHZ }];
## FPGAOL BUTTON & SOFT_CLOCK
set_property -dict { PACKAGE_PIN B18      IOSTANDARD LVCMOS33 }
[get_ports { rst }];
## FPGAOL LED (single-digit-SEGPLAY)
set_property -dict { PACKAGE_PIN C17      IOSTANDARD LVCMOS33 }
[get_ports { counter[0] }];
set_property -dict { PACKAGE_PIN D18      IOSTANDARD LVCMOS33 }
[get_ports { counter[1] }];
set_property -dict { PACKAGE_PIN E18      IOSTANDARD LVCMOS33 }
[get_ports { counter[2] }];
set_property -dict { PACKAGE_PIN G17      IOSTANDARD LVCMOS33 }
[get_ports { counter[3] }];
set_property -dict { PACKAGE_PIN D17      IOSTANDARD LVCMOS33 }
[get_ports { counter[4] }];
set_property -dict { PACKAGE_PIN E17      IOSTANDARD LVCMOS33 }
[get_ports { counter[5] }];
set_property -dict { PACKAGE_PIN F18      IOSTANDARD LVCMOS33 }
[get_ports { counter[6] }];
set_property -dict { PACKAGE_PIN G18      IOSTANDARD LVCMOS33 }
[get_ports { counter[7] }];
```

在FPGAOL实验平台的运行结果如下。（相比于**32**位计数器的情况，**led**闪烁速度较快）

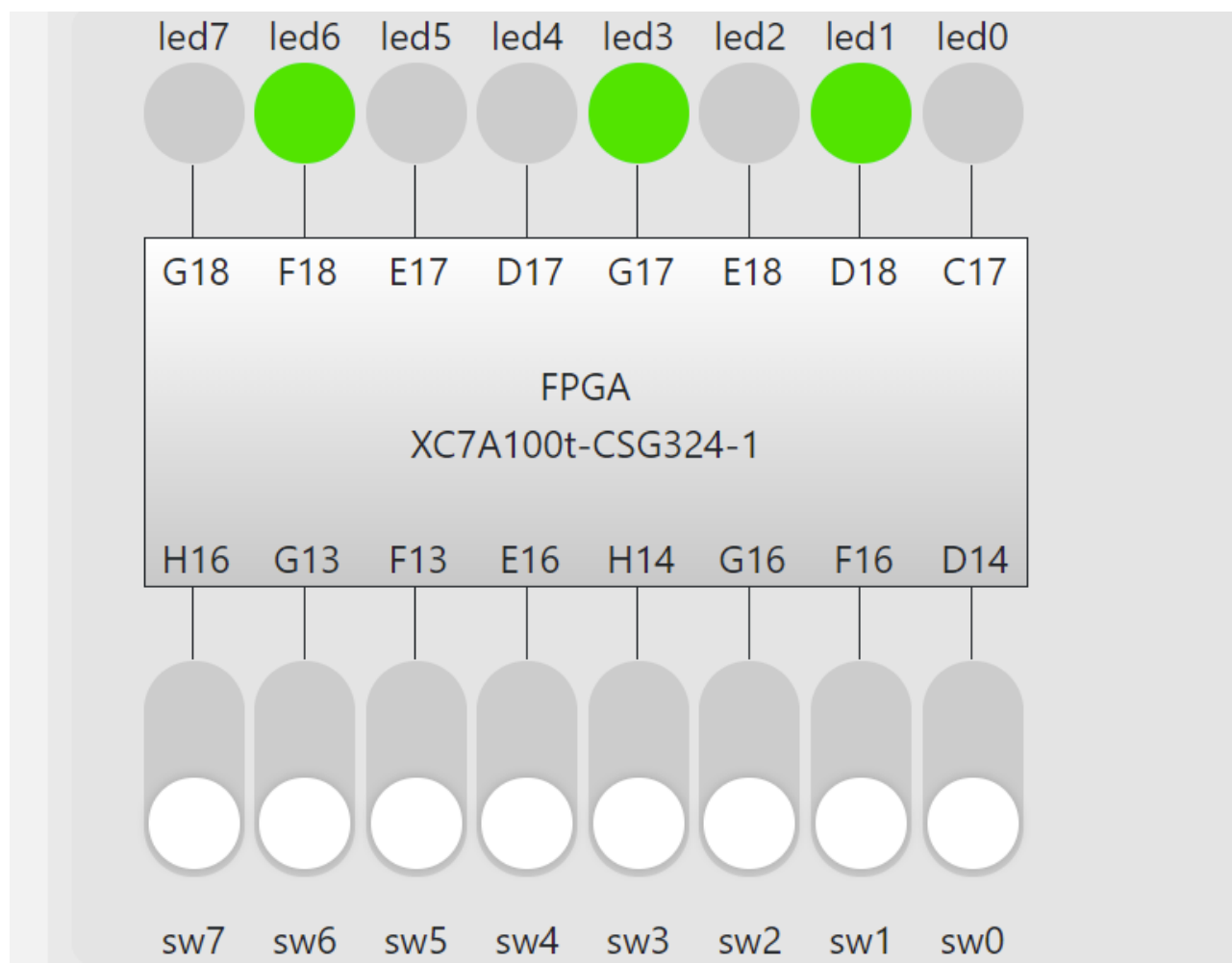


3.2 对于**32**位计数器

32位计数器的编码思路与**30**位计数器类似，实际运行效果的不同表现为**led**闪烁的频率都有明显下降。下面给出**Verilog**代码设计文件，约束文件的内容与**30**位计数器相同（都是将**counter**的7位与7个**led**引脚一一对应）。

```
module test(  
    input clk,  
    input rst,  
    output [7:0] counter  
);  
reg [31:0] temp;  
assign counter = temp[31:24];  
always @(posedge clk or posedge rst) begin  
    if(rst)  
        temp<=1'b0;  
    else  
        temp<=temp+1'b1;  
    end  
endmodule
```


在**FPGAOL**实验平台上烧写后的运行效果如下。



总结与思考

本次实验通过**FPGAOL**实验平台，将实验5的波形图进一步具体化。利用**xdc**文件，将**Verilog**代码中的端口与**FPGA**上的引脚相对应，以完成诸如计数器等一系列的实验设计，更有利于将**Verilog**代码形象化，深入理解**Verilog**语法，增加编写**Verilog**代码的熟练度；同时也初步认识了约束文件，初步了解其功能和编写，为本学期之后的实验及大作业提供了良好的参考与基础。