

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： 黄瑞轩

学生学号： PB20111686

完成日期： 2021.11.19

计算机实验教学中心制

2020 年 09 月

## 实验题目

FPGA 原理及 Vivado 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 Vivado 进行 FPGA 开发的完整流程

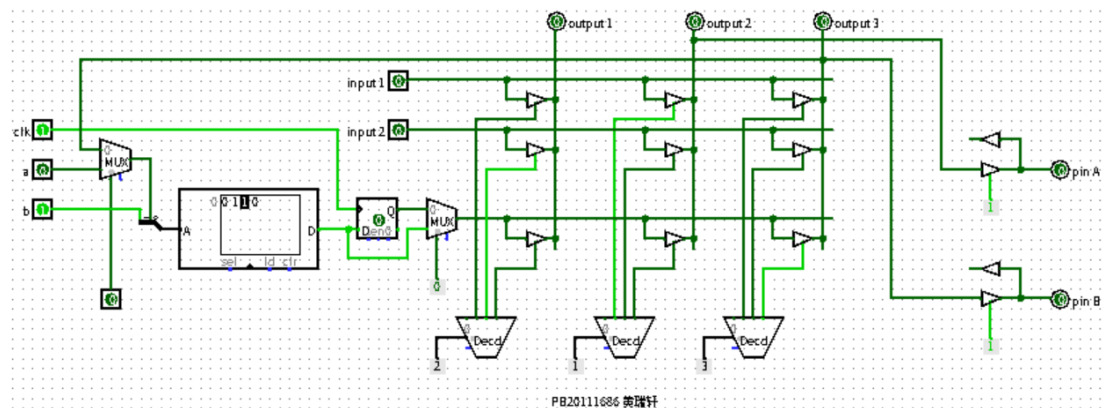
## 实验环境

- VLAB 平台: vlab.ustc.edu.cn
- FPGAOL 实验平台: fpgaol.ustc.edu.cn
- Logisim
- Vivado 工具

## 实验练习

### 【题目 1】

- 首先，根据指导手册中的示例搭建可编辑逻辑单元、交叉互联矩阵以及 IOB 电路，电路图如下。



- 搭建电路时，因为题目要求输出到引脚 B 上，这里改变了译码器的配置数据，将第 3 个输出端移到了输出到引脚 B 的那一个阵列，并且将 output 3 的数据拖回第一个（和 a 并列的）数据选择器；题目要求实现  $a \leftarrow a \oplus b1$ ，因此将 b 的值设置为 1'b1。
- 配置数据如图所示，首先根据  $a \oplus b$  的真值表修改 RAM 的值为 0110，这里要求实现时序电路，因此从左往右第一个 MUX 的 sel 端口应置为 0，以将每次反馈的值作为 a 的新输入；由于是时序电路故第二个 MUX 的 sel 端口应置为 0。最终的配置如上图所示。

## 【题目 2】

- 注意到开关管脚和 LED 管脚的对应关系，这里将所有 LED 对应的端口反过来。修改后的 XDC 文件如下图所示。

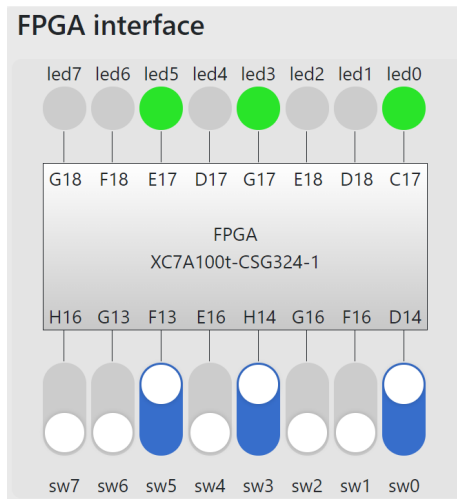
```
set_property -dict {PACKAGE_PIN E3 IOSTANDARD LVCMOS33} [get_ports {clk}];

set_property -dict {PACKAGE_PIN B18 IOSTANDARD LVCMOS33} [get_ports {rst}];

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]}];

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

- 按实验指导手册上的实验过程生成.bit 文件，上传到 fpgaol 平台烧写，结果如下所示。



### 【题目 3】

- 用寄存器存储的 `ctr` 信号作为 `led` 的控制信号，由于要求 `led` 显示高八位，因此编写的设计文件 `top.v` 如下所示。

```
module test(input clk, output reg [7:0] led);  
    reg [29:0] ctr = 0;  
    always@(posedge clk) ctr <= ctr + 1;  
    always@(posedge clk)  
        led <= {ctr[29],ctr[28],ctr[27],ctr[26],ctr[25],ctr[24],ctr[23],ctr[22]};  
endmodule
```

- 与指导手册中的约束文件相比，这里不需要手动输入 `sw` 信号，因此约束文件 `top.xdc` 如下所示。

```
set_property -dict {PACKAGE_PIN E3 IOSTANDARD LVCMOS33} [get_ports {clk}];  
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]}};
```

- 按实验指导手册上的实验过程生成 `.bit` 文件，上传到 `fpgaol` 平台烧写，结果观察到 LED 按计数器模式变动，LED 亮表示该数位的数码为 1，反之则为 0。  
(由于结果是动态的，本报告不方便呈现，已在实验检查阶段呈现，故在这里略去)
- 要将计数器改成 32 位的，只需对设计文件修改如下，波浪线处指示了相比 30 位计数器的设计文件的改动处。

```
module test(input clk, output reg [7:0] led);  
    reg [31:0] ctr = 0;  
    always@(posedge clk) ctr <= ctr + 1;  
    always@(posedge clk)  
        led <= {ctr[31],ctr[30],ctr[29],ctr[28],ctr[27],ctr[26],ctr[25],ctr[24]};  
endmodule
```

- 约束文件不做改动，按实验指导手册上的实验过程生成 `.bit` 文件，上传到 `fpgaol` 平台烧写，结果观察到类似 30 位计数器时的情况，LED 按计数器模式变动，但是 LED 的跳变频率与 30 位时的情况不同。因为此时 `clk` 跳变间隔是不变的，30 位计数器 LED 的跳变频率是 21 位进位，而 32 位计数器 LED 的跳变频率是 23 位进位，后者所需的时间周期更长。这里时钟信号保证了两种计数器实际的“单位跳变时间”相同，因此我们才能对比得出上述结论。

## 总结与思考

- 本次实验中我初步认识了 FPGA，以及利用 Vivado 综合 FPGA 的过程，总体来说收获很大；
- 本次实验的指导手册对 FPGA 做了很详细的解释，每个实验步骤也比较清楚，练习题也是对实验过程稍作改动，所以本次实验是比较简单的；
- 本次实验只有 3 小题，任务量不大；但是若对讲义阅读不仔细、编写程序时粗心大意，便易使程序出现各种问题，Vivado 的 Generate Bitstream 每次需要消耗的时间较长，会导致多出很多调试时间；
- 改进建议：  
在实验讲义中介绍一下 fpgaol 平台中各种有用的资料的位置，比如有资料着重介绍了 fpgaol 上七段数码管的接口用法，但是实验讲义没有着重指示出来。