**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

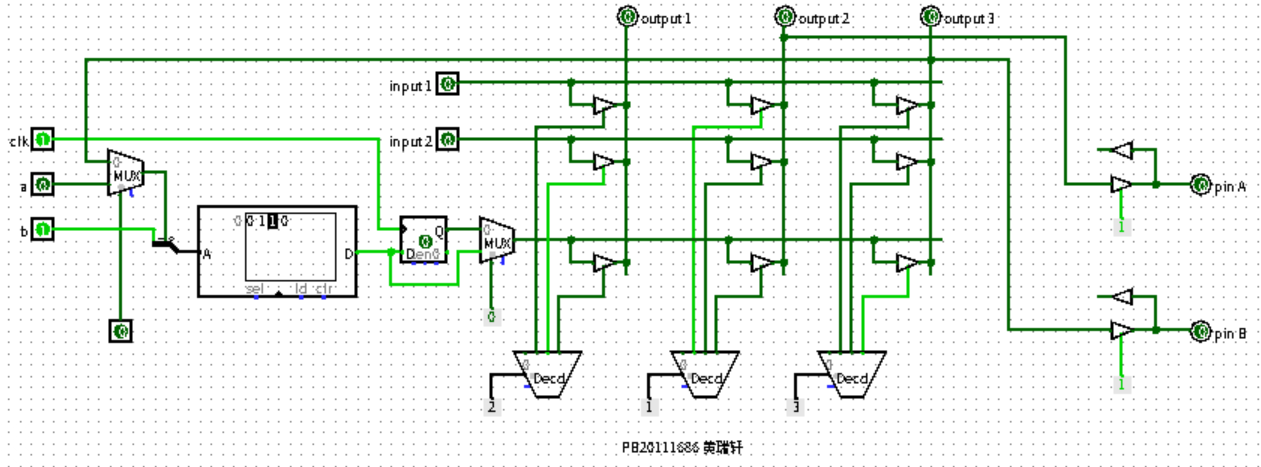
**实验环境**

* 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】**

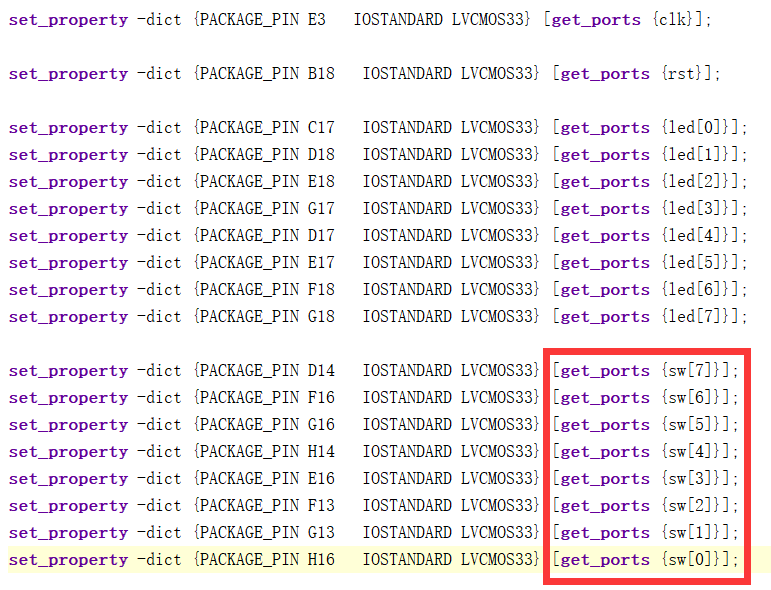
* 首先，根据指导手册中的示例搭建可编辑逻辑单元、交叉互联矩阵以及IOB电路，电路图如下。



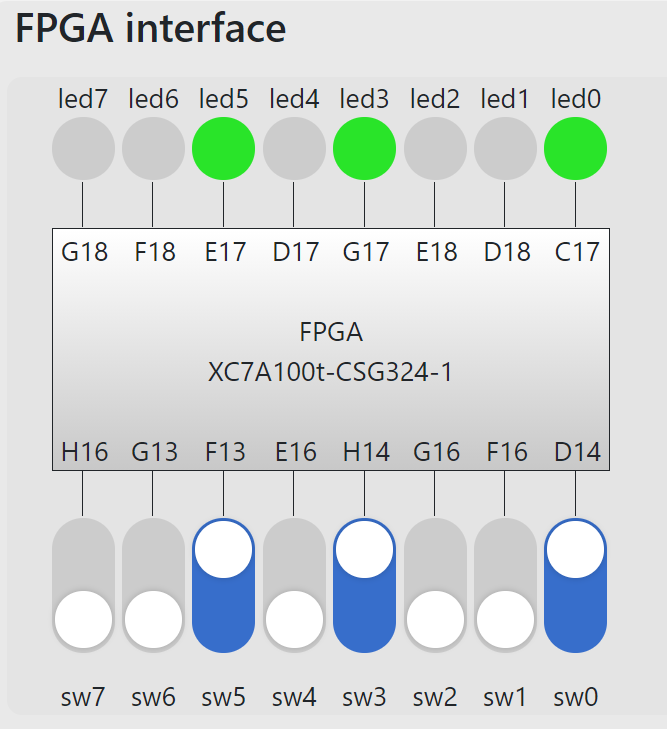
* 搭建电路时，因为题目要求输出到引脚B上，这里改变了译码器的配置数据，将第3个输出端移到了输出到引脚B的那一个阵列，并且将output 3的数据拖回第一个（和a并列的）数据选择器；题目要求实现a <= a ^ 1’b1，因此将b的值设置为1’b1。
* 配置数据如图所示，首先根据a^b的真值表修改RAM的值为0110，这里要求实现时序电路，因此从左往右第一个MUX的sel端口应置为0，以将每次反馈的值作为a的新输入；由于是时序电路故第二个MUX的sel端口应置为0。最终的配置如上图所示。

**【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。



* 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。
* 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;

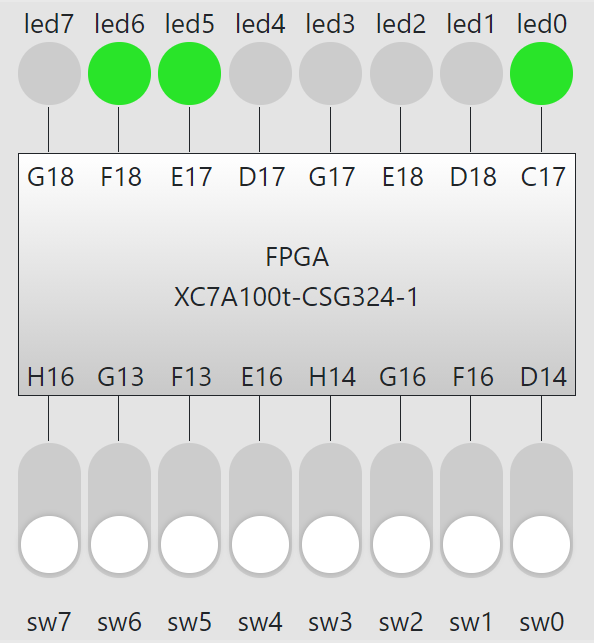
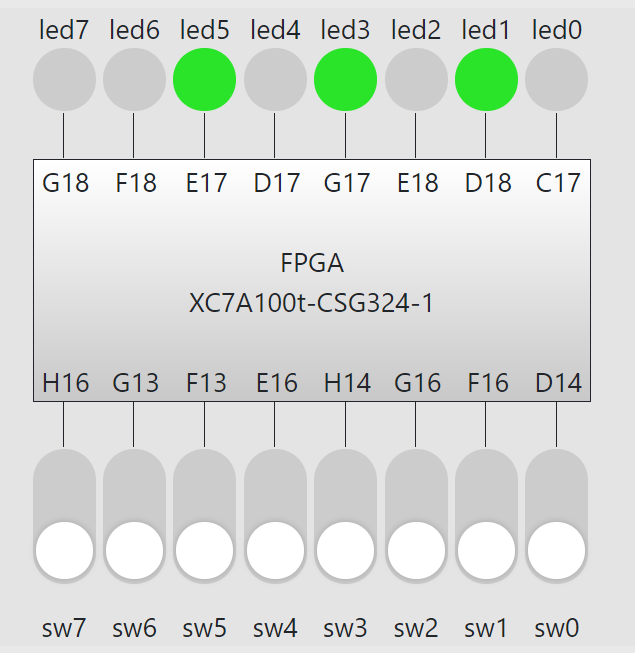
always@(posedge clk) ctr <= ctr + 1;

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。
* 下面是经过相同时间两计数器的截图。可以看到，经过相同时间，30位计数器的高8位比32位计数器的高8位计数的多。

←30位 32位→

**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。