

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：叶子昂

学生学号：PB20020586

完成时间：2021年12月1日

实验题目

FPGA原理及Vivado综合

实验目的

- 了解FPGA的工作原理
- 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开发的完整的流程

实验环境

- Vlab平台
- FPGAOL实验平台
- Logisim工具
- Vivado工具

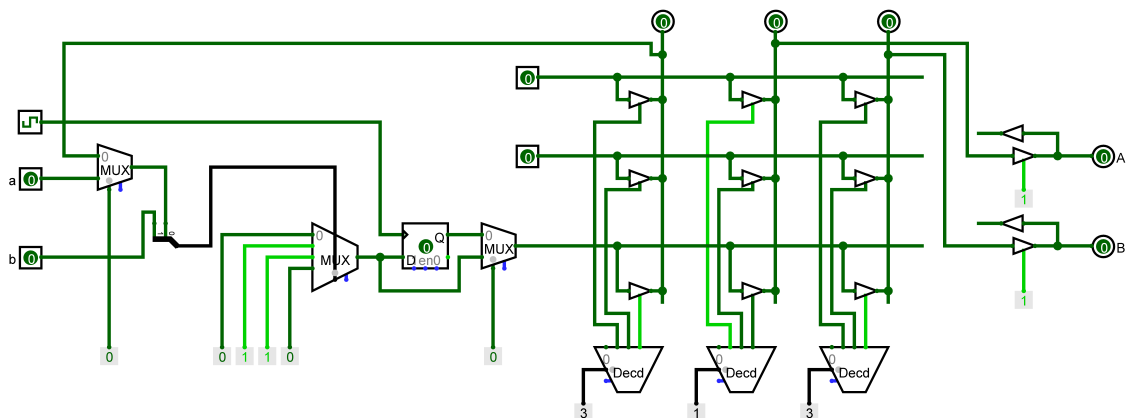
实验练习

题目一：

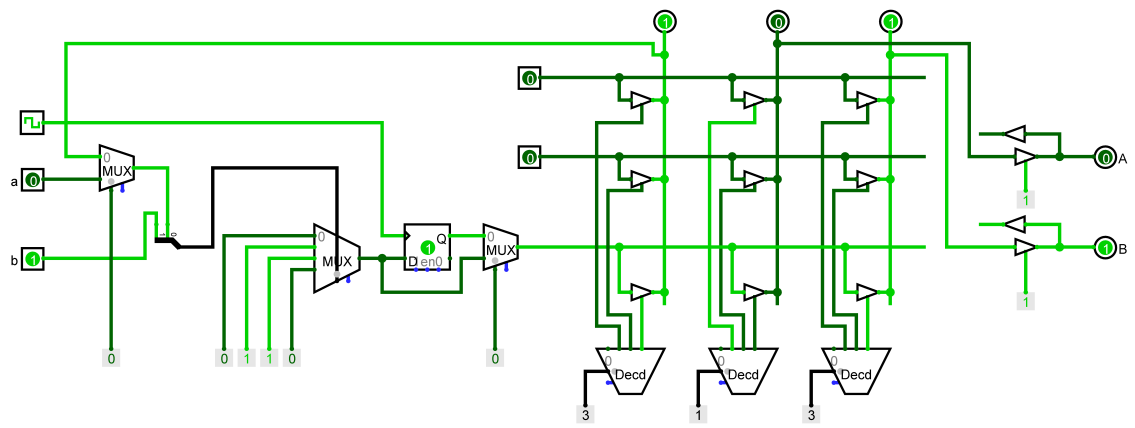
请通过实验中给出的可编程逻辑单元、交叉互连矩阵及 IOB电路图，实现如下代码，并将其输出到引脚 B 上。给出配置数据和电路截图。

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

- 首先绘制出可编程逻辑单元，交叉互连矩阵及IOB电路图。



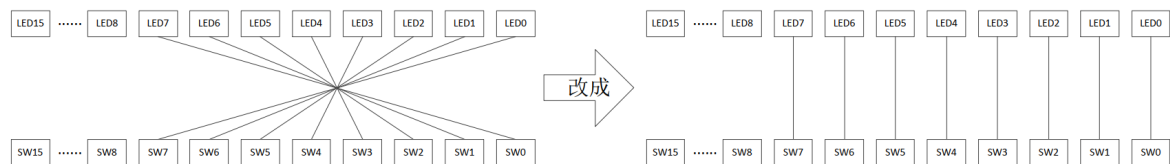
- 功能模块要求使用output reg a，故二选一数据选择器选0端。时钟上升沿时执行异或操作，故四选一数据选择器置为0110，b输入引脚为1。模块实现的为时序电路故第二个二选一数据选择器选0端。由于需要反馈结果和输出结果到引脚B，故可设置交叉互连矩阵为3，1，3。



编程配置为：| 0 | 0110 | 0 | 3 1 3 |

题目二：

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。

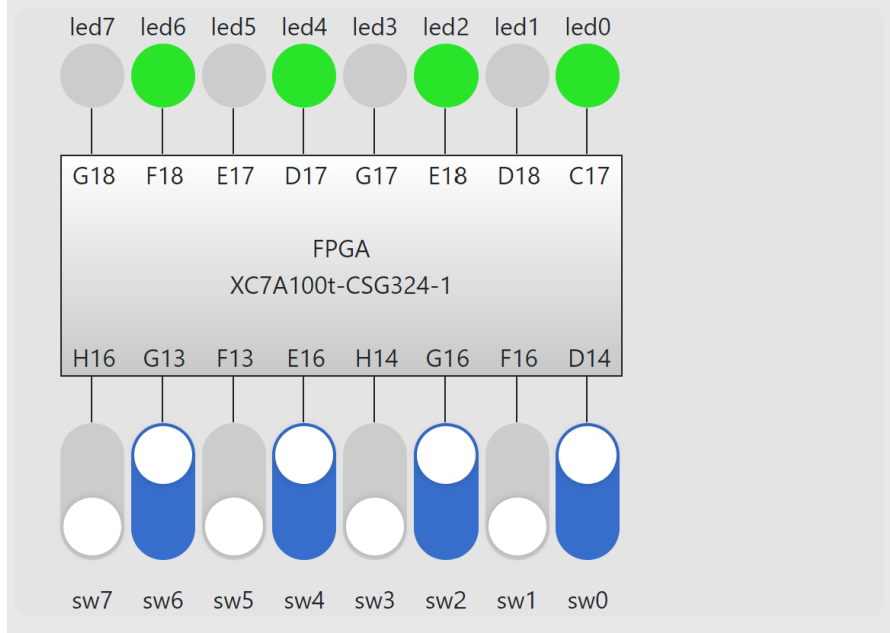


- 先将原文件综合出的bit文件烧写进入FPGA在线平台。发现灯与开关对应关系如上左图所示。
- 打开test.xdc文件，调换输出接口与FPGA端口的对应关系，再综合生成bit文件烧写入FPGA在线平台。
- 部分对应接口如下：

```
set_property -dict { PACKAGE_PIN C17   IOSTANDARD LVCMOS33 } [get_ports {
  led[7] }];
set_property -dict { PACKAGE_PIN D18   IOSTANDARD LVCMOS33 } [get_ports {
  led[6] }];
set_property -dict { PACKAGE_PIN E18   IOSTANDARD LVCMOS33 } [get_ports {
  led[5] }];
set_property -dict { PACKAGE_PIN G17   IOSTANDARD LVCMOS33 } [get_ports {
  led[4] }];
set_property -dict { PACKAGE_PIN D17   IOSTANDARD LVCMOS33 } [get_ports {
  led[3] }];
set_property -dict { PACKAGE_PIN E17   IOSTANDARD LVCMOS33 } [get_ports {
  led[2] }];
set_property -dict { PACKAGE_PIN F18   IOSTANDARD LVCMOS33 } [get_ports {
  led[1] }];
set_property -dict { PACKAGE_PIN G18   IOSTANDARD LVCMOS33 } [get_ports {
  led[0] }];
```

- 测试开关与LED等对应关系如下图：

FPGA interface



修改正确。

题目三:

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。

- 设计 30 位计数器:

```
module conculator (
    input clk,rst,
    output [7:0] cout
);
    reg [29:0] Q;
    always @(posedge clk or posedge rst)
    begin
        if(rst==1)Q<=30'b0;
        else Q<=Q+30'b1;
    end
    assign cout=Q[29:22];
endmodule
```

- 在.xdc中约束管脚，将cout分配到FPGA8个LED端口。利用Vivado综合生成bit文件。

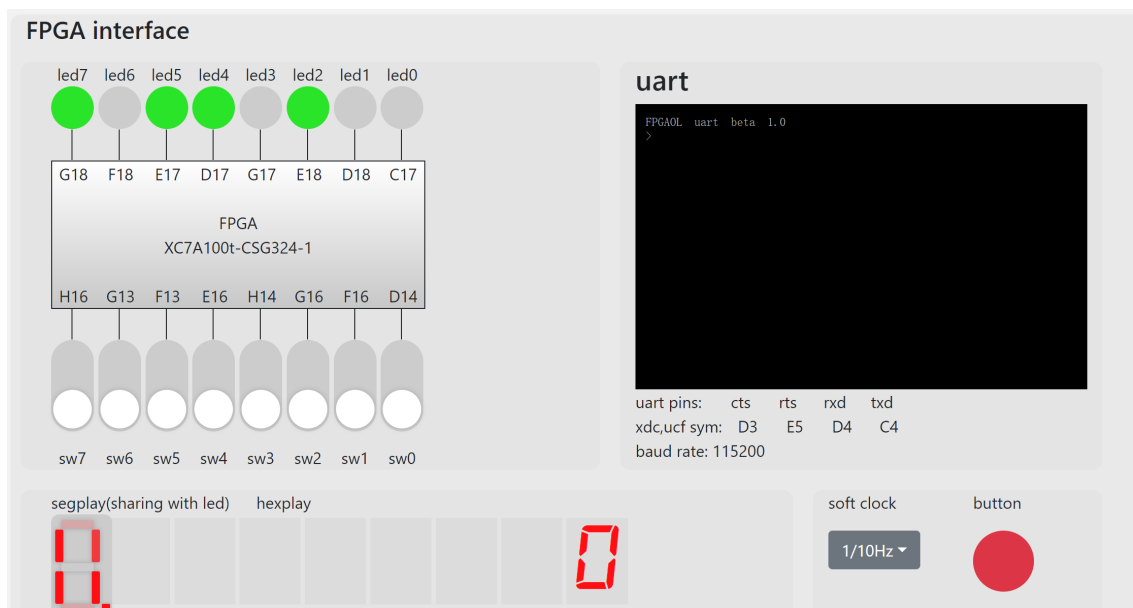
```
## Clock signal
set_property -dict { PACKAGE_PIN E3      IOSTANDARD LVCMOS33 } [get_ports {
    clk }]; #IO_L12P_T1_MRCC_35 Sch=clk100mhz
#create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5}
[get_ports {CLK100MHZ}];
## FPGA0L BUTTON & SOFT_CLOCK
set_property -dict { PACKAGE_PIN B18      IOSTANDARD LVCMOS33 } [get_ports {
    rst }];
## FPGA0L LED (single-digit-SEGPLAY)
set_property -dict { PACKAGE_PIN C17      IOSTANDARD LVCMOS33 } [get_ports {
    cout[0] }];
```

```

set_property -dict { PACKAGE_PIN D18   IOSTANDARD LVCMOS33 } [get_ports {
cout[1] }];
set_property -dict { PACKAGE_PIN E18   IOSTANDARD LVCMOS33 } [get_ports {
cout[2] }];
set_property -dict { PACKAGE_PIN G17   IOSTANDARD LVCMOS33 } [get_ports {
cout[3] }];
set_property -dict { PACKAGE_PIN D17   IOSTANDARD LVCMOS33 } [get_ports {
cout[4] }];
set_property -dict { PACKAGE_PIN E17   IOSTANDARD LVCMOS33 } [get_ports {
cout[5] }];
set_property -dict { PACKAGE_PIN F18   IOSTANDARD LVCMOS33 } [get_ports {
cout[6] }];
set_property -dict { PACKAGE_PIN G18   IOSTANDARD LVCMOS33 } [get_ports {
cout[7] }];

```

- 烧写bit文件进入FPGA在线平台。观察结果：



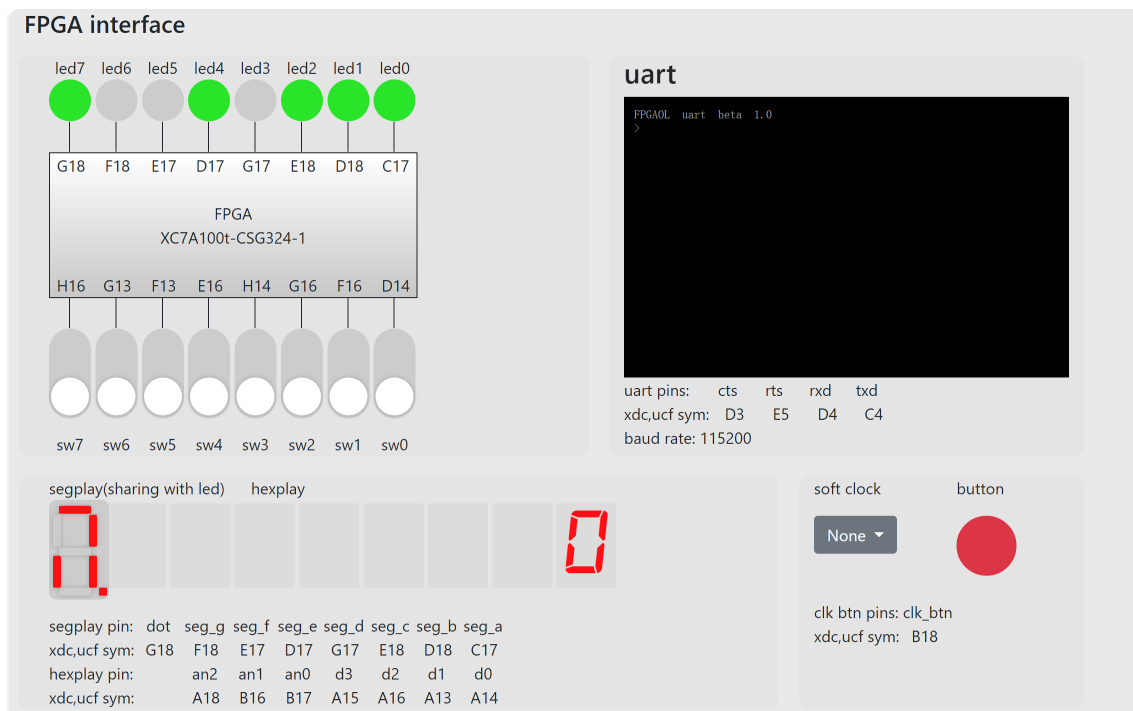
- 设计32位计数器：

```

module conculator (
    input clk,rst,
    output [7:0] cout
);
    reg [31:0] Q;
    always @(posedge clk or posedge rst)
    begin
        if(rst==1)Q<=32'b0;
        else Q<=Q+32'b1;
    end
    assign cout=Q[31:24];
endmodule

```

- 使用相同的.xdc文件，利用Vivado生成bit文件。
- 烧写进入FPGA在线平台，观察结果如下：



- 对比观察发现32位计数器LED变更明显慢于30位计数器。时钟信号起驱动计数器计数的作用（时钟上升沿到来一次计数器加一）。

总结与思考

- 本次实验的收获
 - 通过本次实验，我了解了FPGA的工作原理，知道Verilog约束文件在开发中起的作用。
 - 能够利用所给出的约束文件做出简单的修改并运用在自己的项目中。
 - 学会了利用Vivado进行一套完整的开发流程。
- 本次实验的难易程度
 - 本次实验主要在改.xdc文件上较为简单。
- 本次实验的任务量
 - 本次实验代码部分较为简单，主要工作量在更改约束文件及Vivado综合生成bitstream文件上。任务量适中。
- 对本次实验的建议
 - 本次实验较好的锻炼了我阅读并使用.xdc文件的能力。不过希望能增加一些介绍如何使用hexplay端口的部分。