

实验06 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】

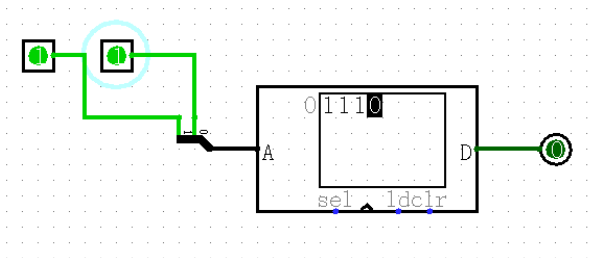
- VLAB 平台：< www.vlab.ustc.edu.cn >
- FPGAOL 实验平台：< www.fpgaol.ustc.edu.cn >
- Logisim
- Vivado 工具

【实验步骤】

Step 3 可编程逻辑单元

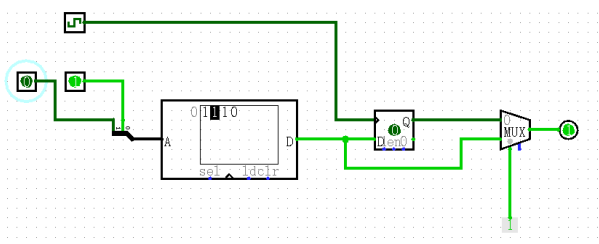
按照指导手册的指示搭建如下电路，发现：

输出端（读端口）显示的数据始终是 RAM 内以输入端（地址）信号为编号的数值，通过改变输入端的值，便可以立即访问到存储在 RAM 内部任意地址的数据，这就是为什么称其为随机访问存储器。



不难发现，通过修改 RAM 的内容，便可以实现所有的两输入组合逻辑。如 `assign o = fun(a,b);` 格式的电路功能都可以实现。

在 RAM 的输出后面添加一个触发器和选择器，并添加一个时钟信号，如下图所示，便实现了对组合逻辑和时序逻辑的支持，通过选择器选择输出信号是否被寄存。



通过配置 RAM 的内容和选择器的选择信号，以下两种语法格式的电路都可以支持：

```
1) assign o = fun (a,b); //组合逻辑
2) always@(posedge clk) //时序逻辑
    o <= fun (a,b);
```

Step 4~6

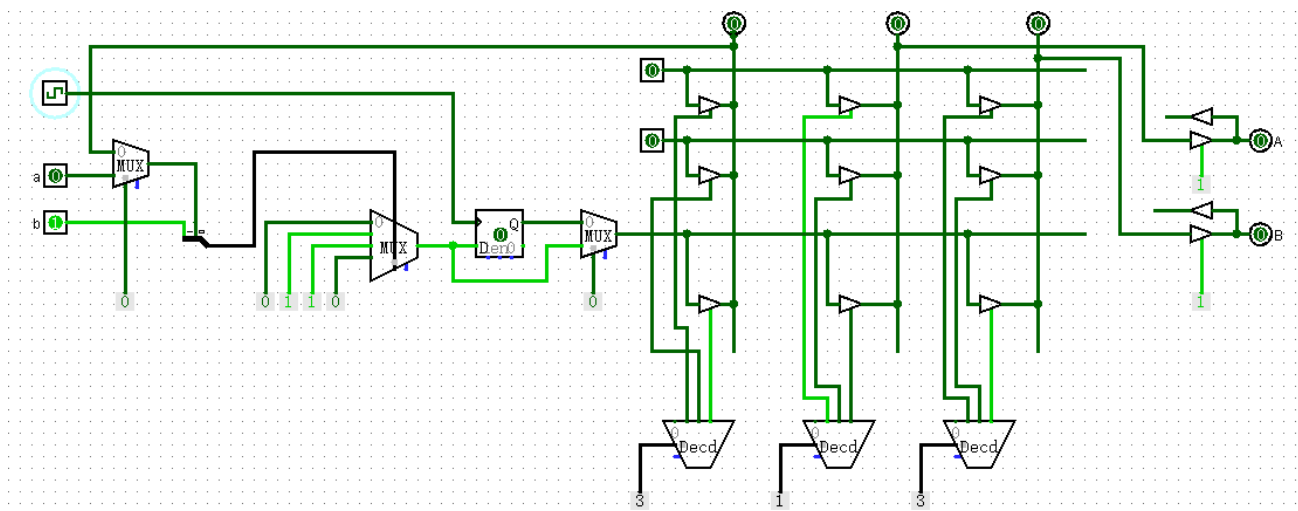
依照实验指导书的要求熟悉交叉互连矩阵的构建，烧写 FPGA 等步骤。在此不再赘述。

【实验练习】

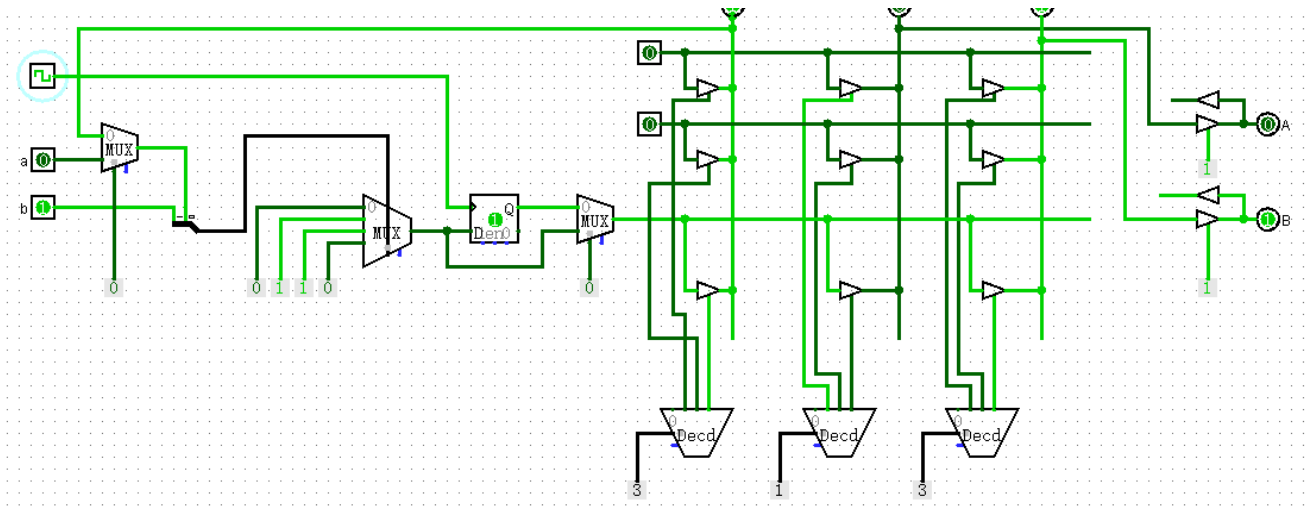
题目 1. 请通过实验中给出的可编程逻辑单元、交叉互连矩阵及 IOB 电路图，实现如下代码，并将其输出到引脚 B 上。给出配置数据和电路截图。

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

首先绘制出可编程逻辑单元，交叉互连矩阵，IOB 电路图如下：

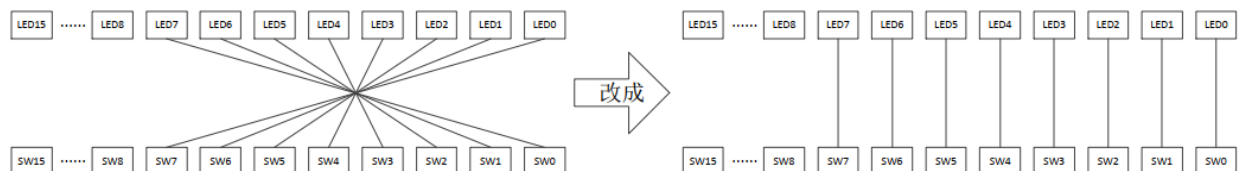


模块要求使用 output reg a, 故二选一数据选择器选0端。时钟上升沿时执行异或操作, 故四选一数据选择器置为0110, b输入引脚为1。模块实现的为时序电路, 故第二个二选一数据选择器选0端。由于需要反馈结果和输出结果到引脚B, 故可设置交叉互联矩阵为3, 1, 3。



编程配置为：| 0 | 0100 | 0 | 313 |

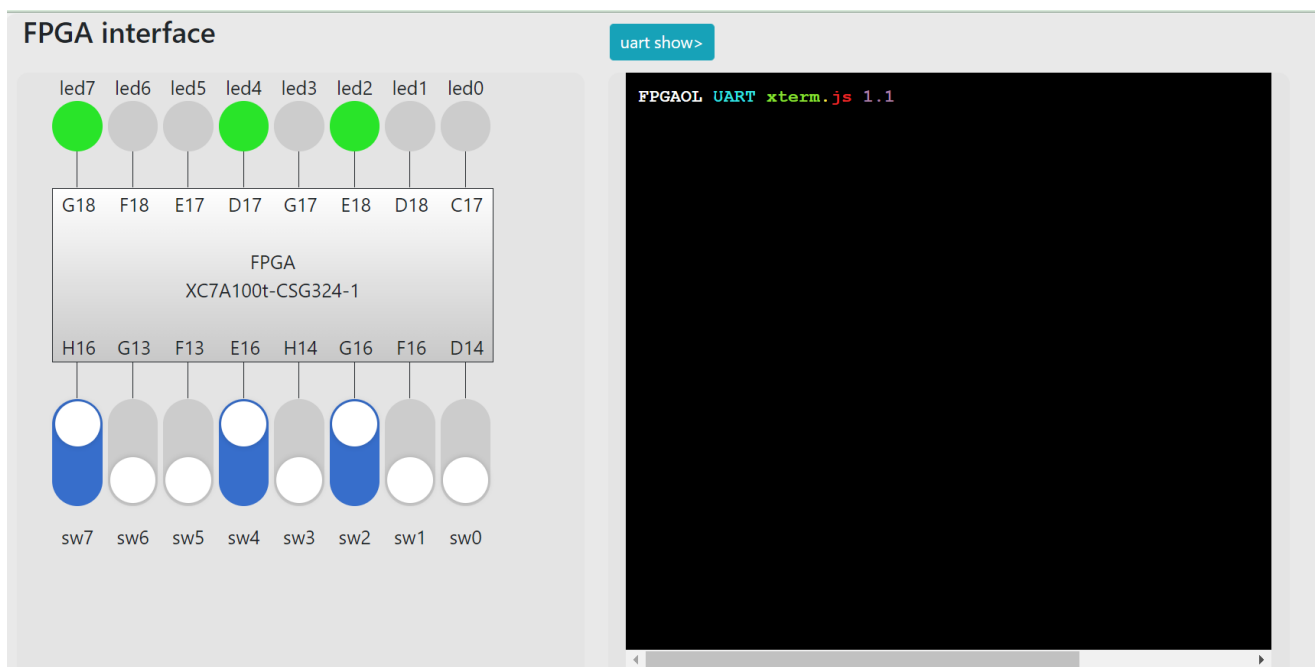
题目 2. 实验中的开关和 LED 的对应关系是相反的，即最左侧的开关控制最右侧的 LED，最右侧的开关控制最左侧的 LED，请修改实验中给出的 XDC 文件，使开关和 LED 一一对应（最左侧的开关控制最左侧的 LED），如下图所示。



调换原 fpgaol.xdc 文件里输出接口与 FPGA 端口的对应关系如下：

```
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] }];
```

将修改后文件综合出的 bit 文件烧写进 FPGA 在线平台，测试开关与 LED 灯的对应关系如下：



可知，修改正确。

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。

设计30位计数器：

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

在 .xdc 中约束脚管，将 cout 分配到8个 LED 端口，并生成 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}];
```

```

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] }];

```

设计32位计数器：

```

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

```

用同样的 .xdc 文件，生成 bit 文件。

把两次生成的 bit 文件烧写进 FPGA 平台，通过观察得出结论：

32位 LED 变更明显慢于30位，时钟信号起驱动计数器计数的作用，每次上升沿来临，计数器加一。

【总结与思考】

1. 请总结本次实验的收获

1. 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。
2. 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。
3. 学会了利用Vivado进行一套完整的开发流程。

2. 请评价本次实验的难易程度

本次实验难度适中，主要难在学习掌握交叉互连矩阵等新出现的概念。

3. 请评价本次实验的任务量

任务量适中。

4. 请为本次实验提供改进建议

对于可编程逻辑单元、交叉互连矩阵的介绍可更为详细，应该提前告知 .xdc 文件的下载位置，避免在错误地敲 .xdc 文件上浪费过多时间。