

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：_____徐奥_____

学生学号：_____PB20061343_____

完成日期：__2021 年 11 月 18 日__

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，实现题目代码，并将其输出到引脚 B 上

分析：

(1) $a \leftarrow a \oplus b$; 所以反馈电路接到输入，最左侧的 MUX 的选择信号为 0.
(2) $a \leftarrow a \oplus b$; 所以当 $a=1$ 时，RAM 输出为 0，当 $a=0$ 时，RAM 输出为 1. 所以第二个 MUX (实现 RAM) 输入为 1010.

(3) 因为是时序电路，所以第三个 MUX 接 0，接入 Q 的值.

(4) 选用第三个译码器，输入为 $2'b11$, 即选择第三行数据输出，接引脚 B

电路截图如图 1.

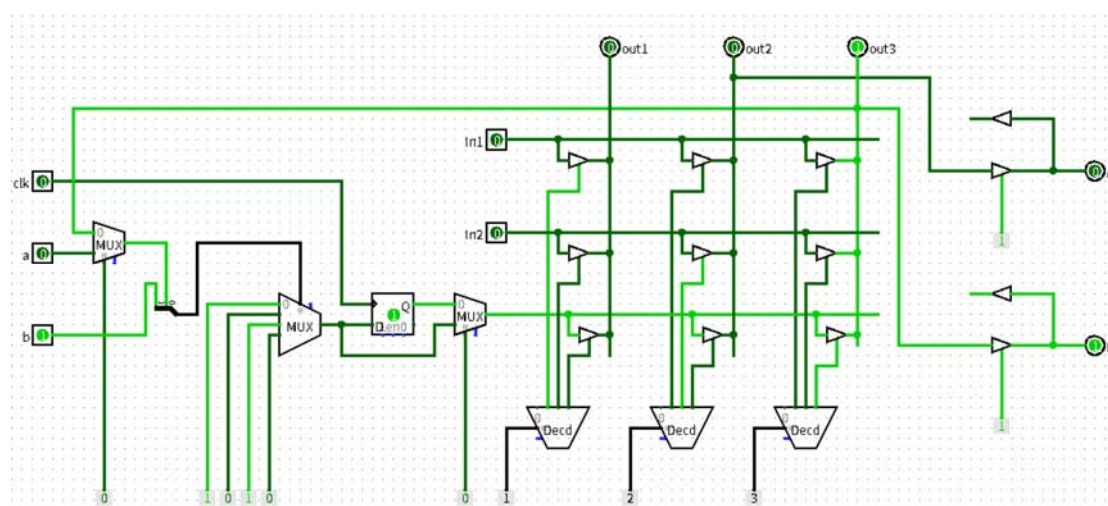


图 1

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）

只需修改 led[7:0] 的顺序，如图 2，原 led 的顺序从上到下依次为 0~7，现在修改为 7~0.

```
## FPGAOL LED (signle-digit-SEGPLAY)
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] }];
```

图 2

题目 3.

(1) 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。

Verilog 代码如下：

```
module Q3_1_v(
    input clk,
    input rst,
    output reg [7:0] led
);

reg [29:0] q;

always @(posedge clk or posedge rst) begin
    if(rst)
        q <= 30'b0;
    else
        q <= q + 30'b1;
end

always @(posedge clk or posedge rst) begin
    if(rst)
        led <= 8'b0;
    else
        led <= {q[29],q[28],q[27],q[26],q[25],q[24],q[23],q[22]};
end

endmodule
```

图 3

引脚约束文件采用第二问修改后的文件，注释掉其中开关部分，如图

```

## 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[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] }];

```

图 4

在 FPGAOL 平台上运行，由于引脚约束文件的设置，在平台上的 led0 为最高位，运行截图如下

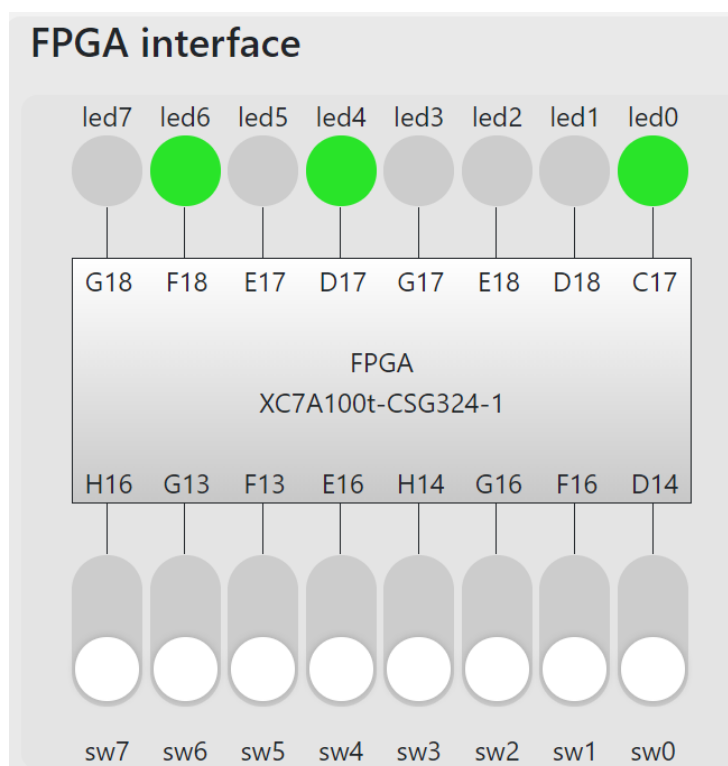


图 5

(2) 将该计数器改成 32 位，将高 8 位输出到 LED

Verilog 代码如下：

```

module Q3_2_v(
    input clk,
    input rst,
    output reg [7:0] led
);

reg [31:0] q;

always @(posedge clk or posedge rst) begin
    if(rst)
        q <= 32'b0;
    else
        q <= q + 32'b1;
end

always @(posedge clk or posedge rst) begin
    if(rst)
        led <= 8'b0;
    else
        led <= {q[31],q[30],q[29],q[28],q[27],q[26],q[25],q[24]};
end

endmodule

```

图 6

引脚约束文件与 30 位计数器的相同。

在 FPGA 平台上运行后发现，30 位计数器 led 的频率是 32 位计数器 led 频率的 4 倍。时钟信号提供了等长的单位时间，每个时钟周期，计数器的最低位加 1。

【总结与思考】

1. 在本次实验中，我学习了如何使用引脚约束文件，学会了如何建立起 Verilog 文件与实际 FPGA 芯片之间的关联。
2. 通过使用 Logisim 软件搭建简单的可编程逻辑单元、交叉互联矩阵、IOB，深入理解了 FPGA 的原理和构成
3. 本次实验难度适中，实验分别使用了了 Logisim 和 Vivado，不同题目之间有所关联
4. 本次实验任务量适中，题目难度梯度上升，循序渐进，建议保持。