

2020-9-13

## 数字电路实验一 选择器

秦嘉余

191220088

1348288404@qq.com

# 目录

---

1.实验目的 .....	1
2.实验原理 .....	1
3.实验环境/器材 .....	1
4.程序代码 .....	1
5.实验步骤 .....	2
6. 测试方法 .....	5
7. 实验结果 .....	5
8. 实验中遇到的问题及解决办法 .....	6
9. 实验得到的启示 .....	6
10.意见和建议 .....	6

## 1.实验目的

---

本次实验的目的是熟悉各种选择器的结构，实验种已经给出了 2 选 1 多路选择器，4 选 1 多路选择器的大致结构，需要自己完整完成一个 **2 位 4 选 1 选择器**。同时在设计这些器件的同时熟悉 Verilog 语言的基本语法与 Quartus 与其组件的基本使用方法。

## 2.实验原理

---

数据选择器是一种常用的数字电路器件，由数据的选择端，输入端与输出端构成，其作用为根据选择端的输入选择输入端的某一位，传送到输出端。这种简单的电路器件，仅需要为数不多的基本门电路，根据真值表画出卡诺图即可由门电路组成。多路选择器则可以由单路选择器级联而成。

## 3.实验环境/器材

---

本次实验的环境为 Quartus17.1 版本

本次实验的器材为 DE10 Standard 开发板

## 4.程序代码

---

本次实验需要实现 **2 位 4 选 1 选择器**，由 4 跟输入端 X0,X1,X2,X3,一个选择端 Y，一个输出端 F,上述的每一个输入输出均为 2 位长度

由于选择器的特性，所以用 case 语句实现最为简单，同时检测任意输入的时钟变化，使用 always 语句，所以应该将 F 定义为 reg 寄存器类型

代码如下：

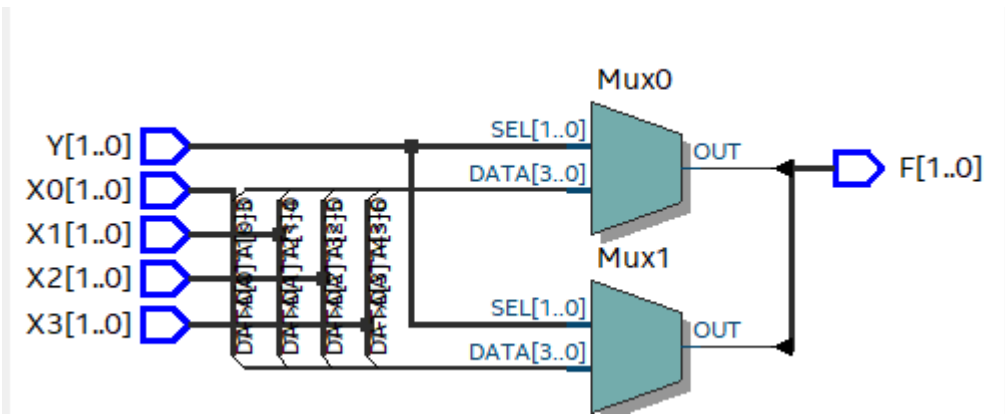
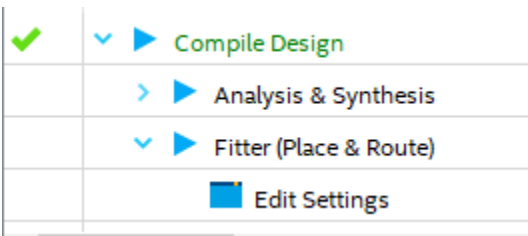
```
module mux_2bits_41(x0,x1,x2,x3,Y,F);
  input [1:0] x0,x1,x2,x3;
  input [1:0] Y;
  output reg [1:0]F;

  always @(*)
  case (Y)
    0: F=x0;
    1: F=x1;
    2: F=x2;
    3: F=x3;
    default: F=2'b00;
  endcase
endmodule
```

## 5.实验步骤

该部分主要展现是实验中各步骤，该实验的步骤较为普通。

**第一步：**根据器件建立合适的项目文件，新建实验代码文件，键入相关代码，并且分析与综合，查看电路原理图，展示如下：

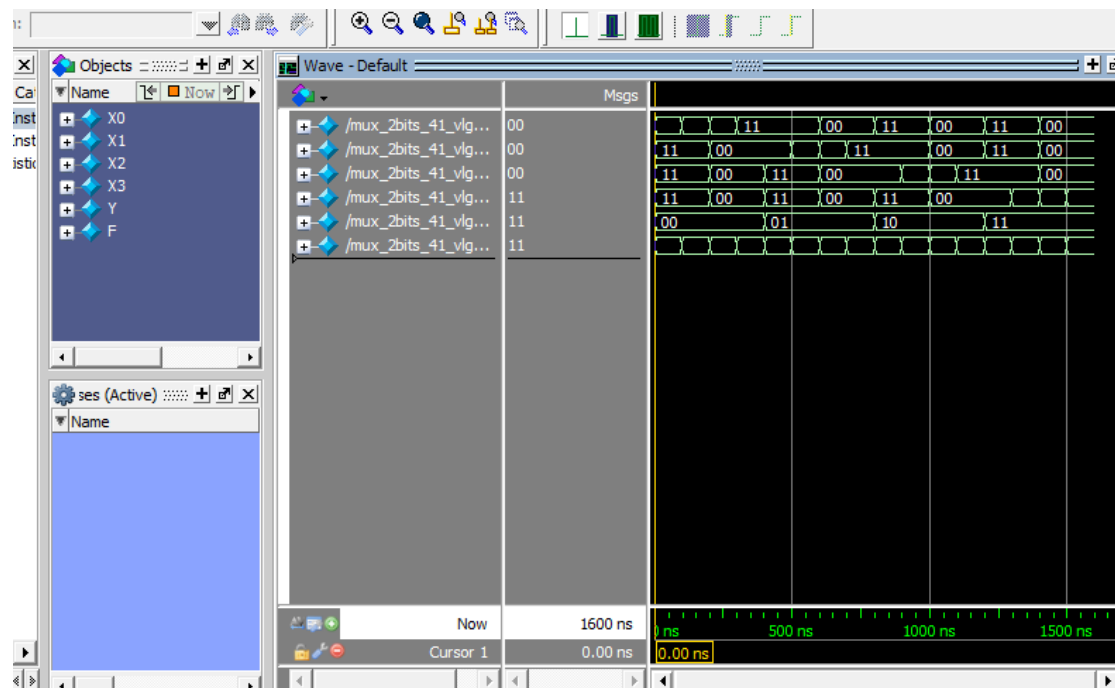


**第二步：**编写模拟测试代码，在 Test Benches 中链接相关模拟测试文件，最后利用 RTL Simualtion 查看波形图，测试程序是否正确，展示如下：

```

initial
begin
    // code that executes only once
    // insert code here --> begin
    Y=2'b00; X0=2'b00; X1=2'b11; X2=2'b11; X3=2'b11;
        X0=2'b01;
        X0=2'b10; X1=2'b00; X2=2'b00; X3=2'b00;
        X0=2'b11;
    Y=2'b01; X0=2'b11; X1=2'b00; X2=2'b11; X3=2'b11;
        X1=2'b01;
        X0=2'b00; X1=2'b10; X2=2'b00; X3=2'b00;
        X1=2'b11;
    Y=2'b10; X0=2'b11; X1=2'b11; X2=2'b00; X3=2'b11;
        X2=2'b01;
        X0=2'b00; X1=2'b00; X2=2'b10; X3=2'b00;
        X2=2'b11;
    Y=2'b11; X0=2'b11; X1=2'b11; X2=2'b11; X3=2'b00;
        X3=2'b01;
        X0=2'b00; X1=2'b00; X2=2'b00; X3=2'b10;
        X3=2'b11;
    end
end

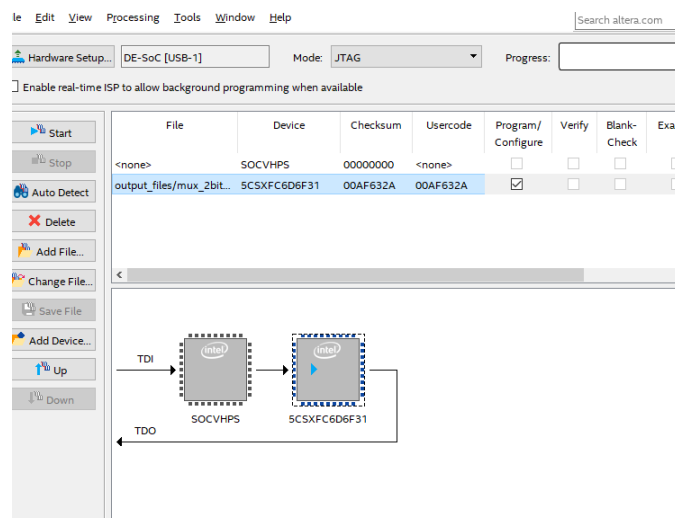
```



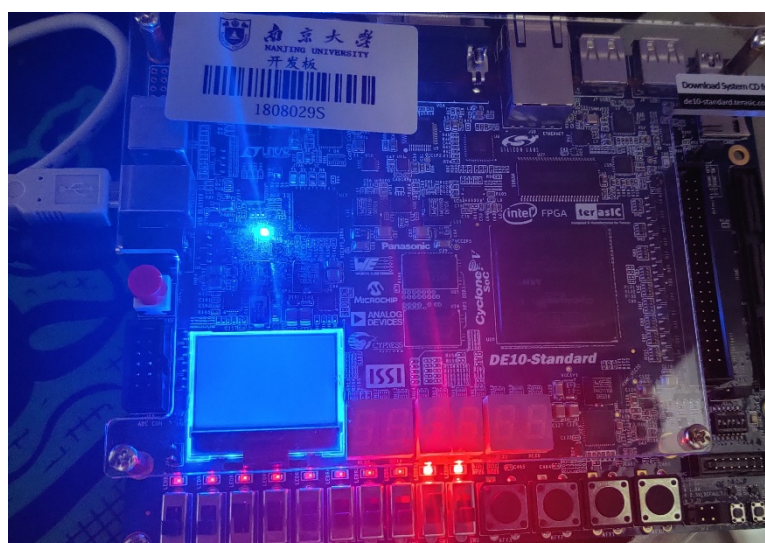
**第三步：**第二步正确后，将引脚对应到开发板的相应接口上，再编译全部项目，连接 DE10 Standard 开发板到电脑，将程序写入开发板上，展示如下：

✓	▼ ▶	Compile Design
✓	> ▶	Analysis & Synthesis
✓	> ▶	Fitter (Place & Route)
✓	> ▶	Assembler (Generate prog
✓	> ▶	TimeQuest Timing Analys
✓	> ▶	EDA Netlist Writer
	🔧	Edit Settings
	🔌	Program Device (Open Program

	Node Name	Direction	Location	I/O Bank
out	F[1]	Output	PIN_AB23	5A
out	F[0]	Output	PIN_AA24	5A
in	X0[1]	Input	PIN_AC30	5B
in	X0[0]	Input	PIN_AB28	5B
in	X1[1]	Input	PIN_V25	5B
in	X1[0]	Input	PIN_W25	5B
in	X2[1]	Input	PIN_AD28	5B



第四步：写入开发板后，测试开发板输入输出是否正确，战术如下：



自此，该实验完成

## 6. 测试方法

本次实验采用测仿真仿真的方法， 编写的测试仿真代码如下：

```
module mux_2bits_41_vlg_tst();  
    reg [1:0] x0;  
    reg [1:0] x1;  
    reg [1:0] x2;  
    reg [1:0] x3;  
    reg [1:0] Y;  
    wire [1:0] F;  
  
    mux_2bits_41 i1 (  
        .F(F),  
        .X0(x0),  
        .X1(x1),  
        .X2(x2),  
        .X3(x3),  
        .Y(Y)  
    );  
    initial  
    begin  
        Y=2'b00; x0=2'b00; x1=2'b11; x2=2'b11; x3=2'b11; #10;  
        x0=2'b01; #10;  
        x0=2'b10; x1=2'b00; x2=2'b00; x3=2'b00; #10;  
        x0=2'b11; #10;  
        Y=2'b01; x0=2'b11; x1=2'b00; x2=2'b11; x3=2'b11; #10;  
        x1=2'b01; #10;  
        x0=2'b00; x1=2'b10; x2=2'b00; x3=2'b00; #10;  
        x1=2'b11; #10;  
        Y=2'b10; x0=2'b11; x1=2'b11; x2=2'b00; x3=2'b11; #10;  
        x2=2'b01; #10;  
        x0=2'b00; x1=2'b00; x2=2'b10; x3=2'b00; #10;  
        x2=2'b11; #10;  
        Y=2'b11; x0=2'b11; x1=2'b11; x2=2'b11; x3=2'b00; #10;  
        x3=2'b01; #10;  
        x0=2'b00; x1=2'b00; x2=2'b00; x3=2'b10; #10;  
        x3=2'b11; #10;  
    end  
endmodule
```

使用的测试数据的选取验证了大部分有意义的情况，选择端选取了全部的 00-11 四种情况，每种情况对应的选择端分别输入 00-11，对于选择端为 00，当 X0 为 00，01 时其他的输入端都为 11，当 X0 为 10，11 时，其他的输入端都为 00。其他的选择端类似，这样大大简化了测试的数据内容，且包含了一切的有意义的输入选择组合

## 7. 实验结果

仿真结果如下：

