

# 数字电路与数字系统实验

---

## 实验一 选择器

---

计算机科学与技术系

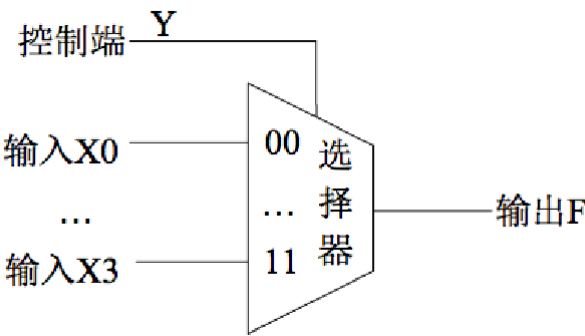
191220008 陈南瞳  
[924690736@qq.com](mailto:924690736@qq.com)

2020.9.12

# 一、实验目的

## 2 位 4 选 1 选择器

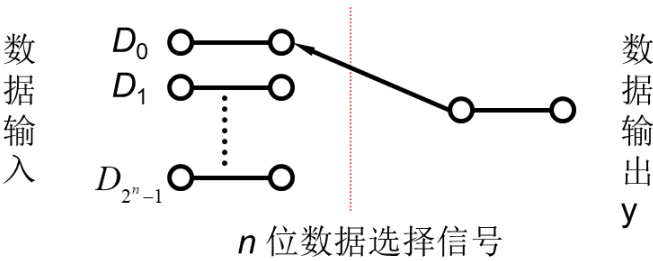
用 case 语句实现一个 2 位 4 选 1 的选择器，选择器有 5 个 2 位输入端，分别为 X0,X1,X2,X3 和 Y，输出端为 F；X0,X1,X2,X3 是四个 2 位的输入变量。输出 F 端受控制端 Y 的控制，选择其中的一个 X 输出，当 Y= 00 时，输出端输出 X0，即 F=X0；当 Y=01 时，输出端输出 X1，即 F=X1； 以此类推。



# 二、实验原理（知识背景）

数据选择器也称为多路转换器，多路复用器或多路开关

2n 数据输入, n 控制信号/选择信号, 1 输出  
用在将 2n 点连接到 1 个点，控制信号是连接到输出的输入信号的二进制编码



$$y = \sum_{i=0}^{2^n-1} m_i D_i$$

输出函数 y，为 n 位选择变量最小项 mi 与上相应数据输入线 Di 之和

常用的数据选择器有 2 选 1、4 选 1、8 选 1、16 选 1 等多种类型

### 三、实验环境/器材等

#### 1) 软件环境:

Quartus (Quartus Prime 17.1) Lite Edition

#### 2) 硬件环境:

DE10-Standard 开发板

FPGA 部分:

Intel Cyclone V SE 5CSXFC6D6 F31C6N

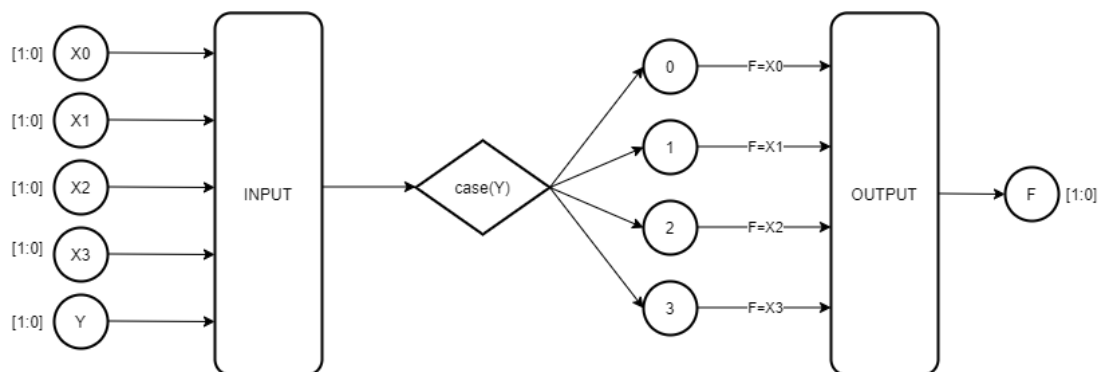
- 110K 逻辑单元
- 5,761Kbit RAM

HPS 部分:

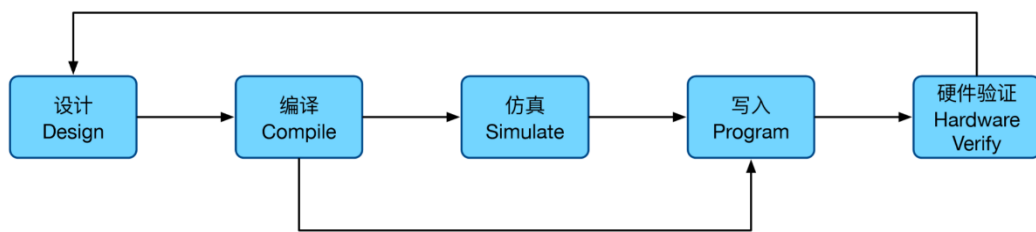
Dual-core ARM Cortex A9

- 925MHz
- 1GB DDR

### 四、程序代码或流程图



### 五、实验步骤/过程



设计：

```

1 module EXP1(X0, X1, X2, X3, Y, F);
2     input [1:0] X0;
3     input [1:0] X1;
4     input [1:0] X2;
5     input [1:0] X3;
6     input [1:0] Y;
7     output reg [1:0] F;
8
9     always @ (X0 or X1 or X2 or X3 or Y)
10        case (Y)
11            0: F = X0;
12            1: F = X1;
13            2: F = X2;
14            3: F = X3;
15            default: F = 2'b00;
16        endcase
17
18 endmodule
19

```

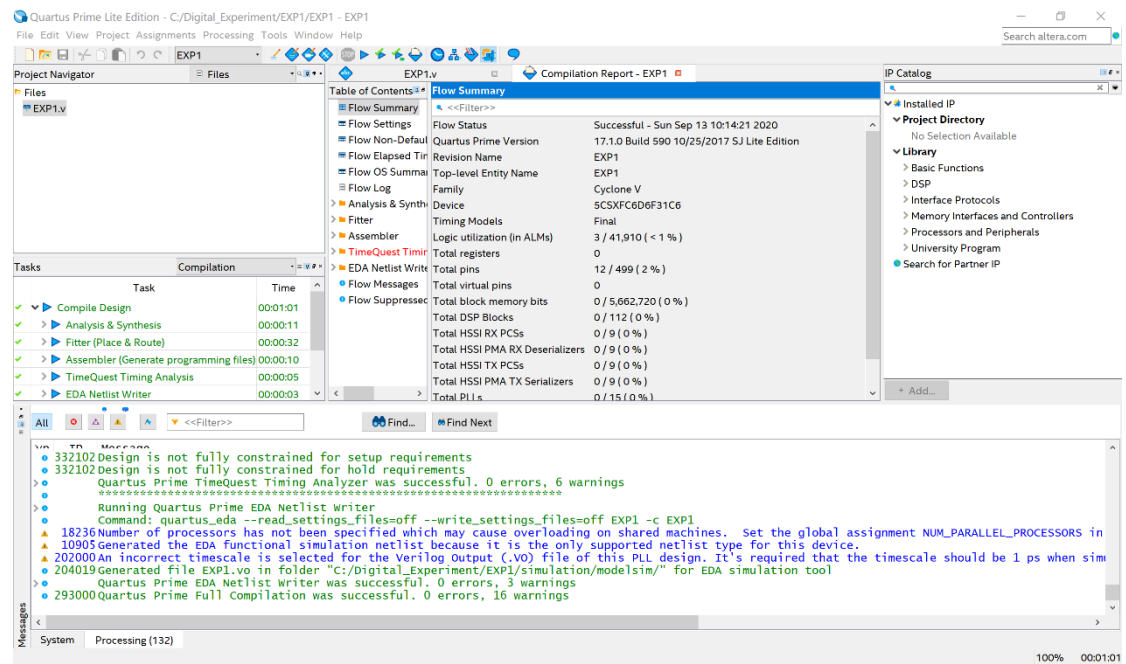
测试：

```

Y = 2'b00; X0 = 2'b00; X1 = 2'b00; X2 = 2'b00; X3 = 2'b00; #10;
X0 = 2'b01; #10;
X0 = 2'b10; #10;
X0 = 2'b11; #10;
Y = 2'b01; X0 = 2'b00; X1 = 2'b00; X2 = 2'b00; X3 = 2'b00; #10;
X1 = 2'b01; #10;
X1 = 2'b10; #10;
X1 = 2'b11; #10;
Y = 2'b10; X0 = 2'b00; X1 = 2'b00; X2 = 2'b00; X3 = 2'b00; #10;
X2 = 2'b01; #10;
X2 = 2'b10; #10;
X2 = 2'b11; #10;
Y = 2'b11; X0 = 2'b00; X1 = 2'b00; X2 = 2'b00; X3 = 2'b00; #10;
X2 = 2'b01; #10;
X2 = 2'b10; #10;
X2 = 2'b11; #10;

```

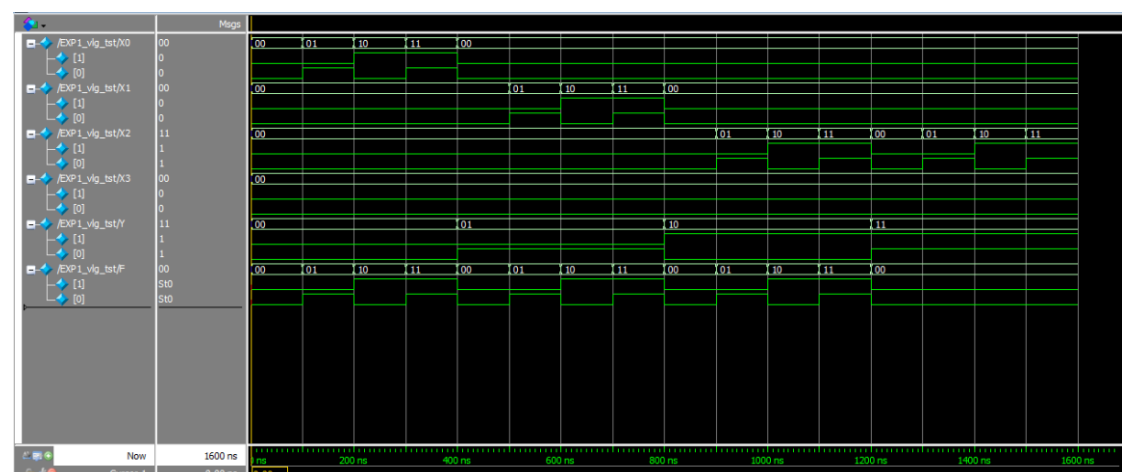
**编译:**



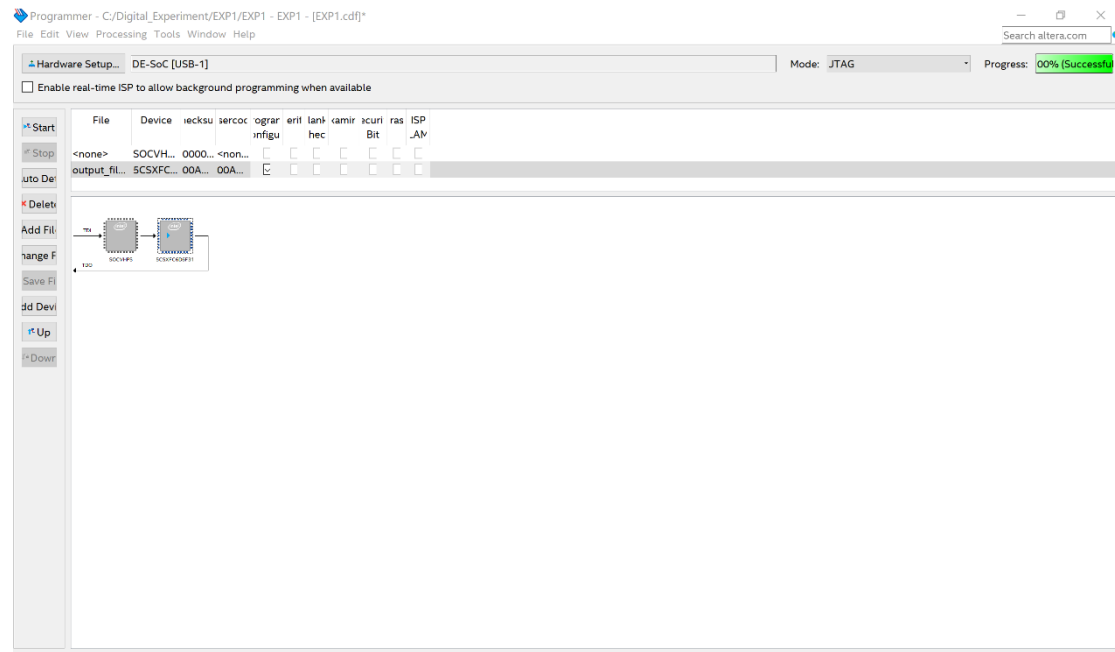
### 引脚分配：

[illegible]

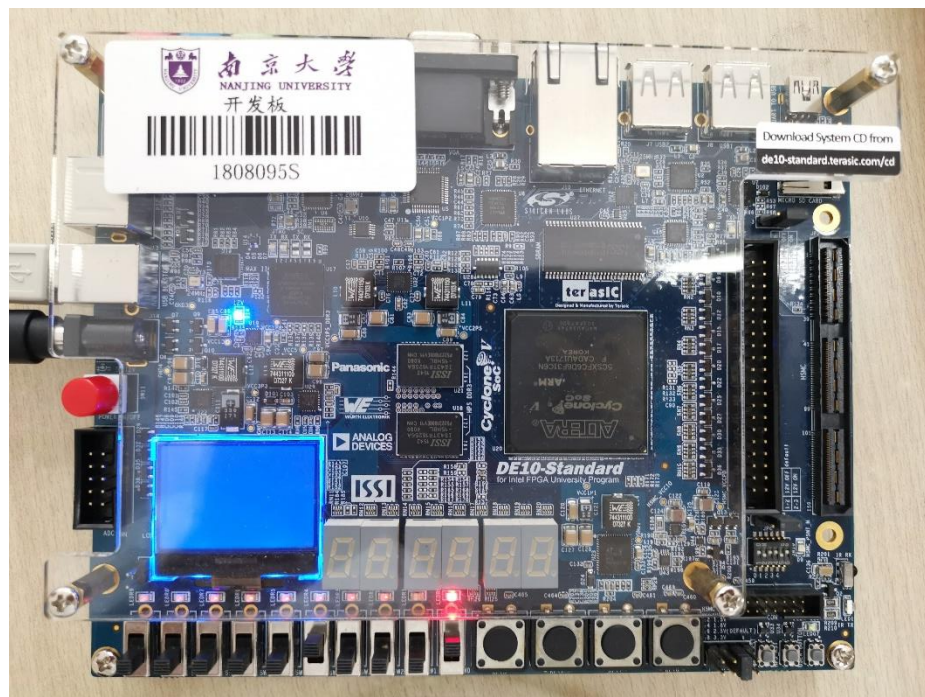
**仿真：**



写入:



### 硬件验证：



# 六、测试方法

## Test Bench

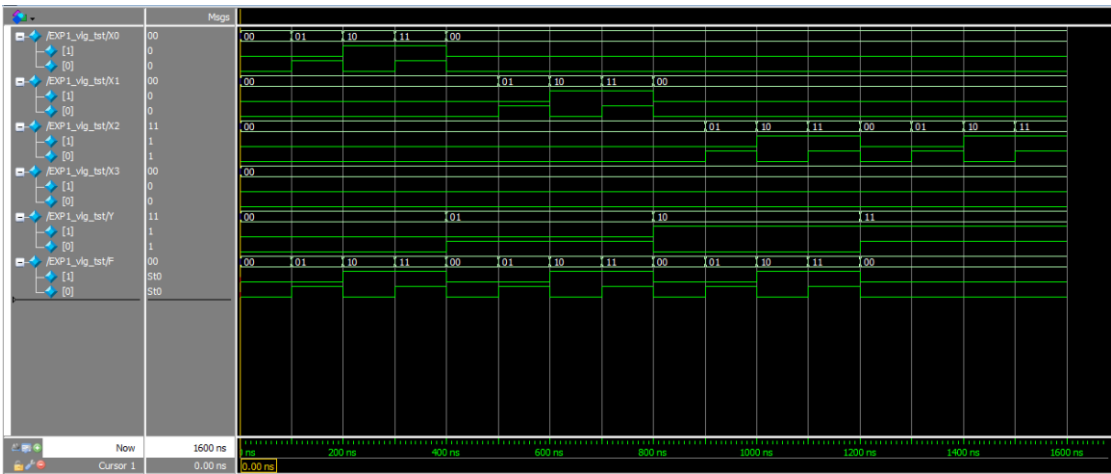
测试当 Y = 0, Y = 1, Y = 2, Y = 3 时,

F 是否分别等于 X0, X1, X2, X3

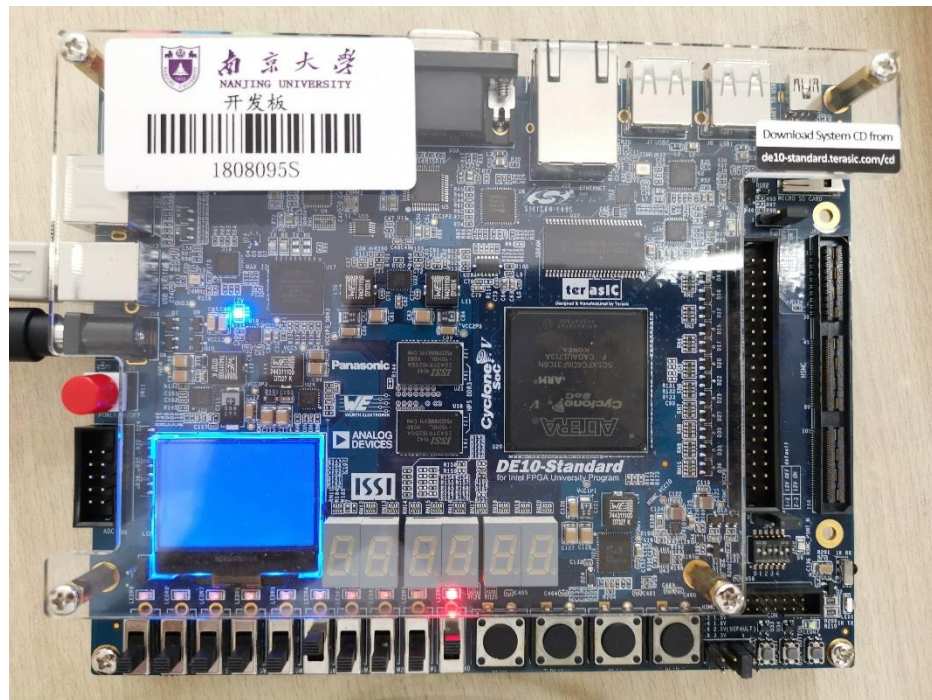
```
Y = 2'b00; X0 = 2'b00; X1 = 2'b00; X2 = 2'b00; X3 = 2'b00; #10;
           X0 = 2'b01; #10;
           X0 = 2'b10; #10;
           X0 = 2'b11; #10;
Y = 2'b01; X0 = 2'b00; X1 = 2'b00; X2 = 2'b00; X3 = 2'b00; #10;
           X1 = 2'b01; #10;
           X1 = 2'b10; #10;
           X1 = 2'b11; #10;
Y = 2'b10; X0 = 2'b00; X1 = 2'b00; X2 = 2'b00; X3 = 2'b00; #10;
           X2 = 2'b01; #10;
           X2 = 2'b10; #10;
           X2 = 2'b11; #10;
Y = 2'b11; X0 = 2'b00; X1 = 2'b00; X2 = 2'b00; X3 = 2'b00; #10;
           X2 = 2'b01; #10;
           X2 = 2'b10; #10;
           X2 = 2'b11; #10;
```

# 七、实验结果

仿真结果：



实际结果：



结论：实际结果与仿真结果完全一致，且与真值表完全一致

## 八、实验中遇到的问题及解决办法

### 1、编译不通过

解决办法：

- ①观察错误提示，对照着自己的代码检查
- ②与示例代码反复对照，观察是否是语法出错

### 2、引脚不知道怎么分配

解决办法：

在 DE10-Standard 使用手册中找到了引脚分配表，对照着引脚分配表进行了合理的分配设置



## 九、实验得到的启示

在遇到陌生的环境和语言时，不需要过度的紧张和手足无措，应当按照步骤一步一步地熟悉新环境，多看示例的代码，联系自己已经掌握的语言和环境的操作方式，即可理解新环境和新语言的操作方式，就能够做到很快的上手

## 十、意见和建议

老师对 DE10-Standard 开发平台和 Verilog 语言的讲解很详细，但在没怎么实践之前，还是一头雾水，第一节课感觉什么都没学到，在自己亲手实践过后，才明白课件中许多东西的意思。可以提前通知同学操作一遍，然后在课上讲解，这样同学们可能更容易理解。