

《数字逻辑》实验报告

姓名	汤英琦	年级	2024 级（大二）
学号	20241280	专业、班级	计算机科学与技术卓越 01
实验名称	实验三 4 选 1 多路选择器		
实验时间	2025. 9. 24	实验地点	D1401
实验成绩		实验性质	<input type="checkbox"/> 验证性 <input type="checkbox"/> 设计性 <input type="checkbox"/> 综合性
<p>教师评价：</p> <p><input type="checkbox"/>算法/实验过程正确； <input type="checkbox"/>源程序/实验内容提交 <input type="checkbox"/>程序结构/实验步骤合理；</p> <p><input type="checkbox"/>实验结果正确； <input type="checkbox"/>语法、语义正确； <input type="checkbox"/>报告规范；</p> <p>评语：</p> <p>评价教师签名（电子签名）：</p>			
<p>一、实验目的</p> <p>熟悉开发环境：学习并熟悉 Vivado 开发环境和 FPGA 开发板的使用</p> <p>掌握基本电路设计：设计并验证基本的门电路（与门、或门、非门等）和多路选择器</p> <p>学习 Verilog HDL 语言：运用 Verilog HDL 语言的数据流描述和行为描述方法进行电路设计</p> <p>掌握 IP 核封装与调用：学习将基本门电路封装为可变数据宽度的 IP 核，并在后续实验中调用</p> <p>掌握仿真与验证：通过仿真、查看 RTL 电路图和下载到开发板等多种方式，验证电路设计的正确性</p> <p>对比不同设计方法：比较使用 Block Design 和添加源代码等不同方法实现相同</p>			

电路的异同

二、实验项目内容

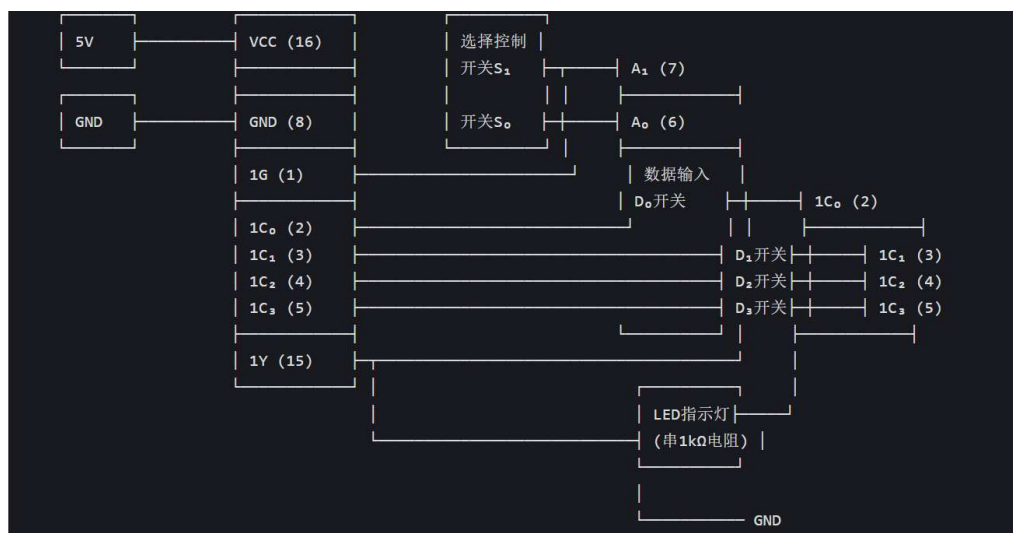
分别使用 Block Design 设计方法和添加源代码的方法设计一个 1 位 4 选 1 多路选择器，并通过仿真、看 RTL 电路图、下载到开发板验证其正确性；同时比较使用不同方法实现相同电路的异同，比较维度包括逻辑资源、RTL 电路图、仿真波形等方面

使用 Verilog HDL 语言的行为描述方法设计一个 3 位数据的 4 选 1 多路选择器 mux4x3，通过仿真、看 RTL 电路图、下载到开发板验证其正确性

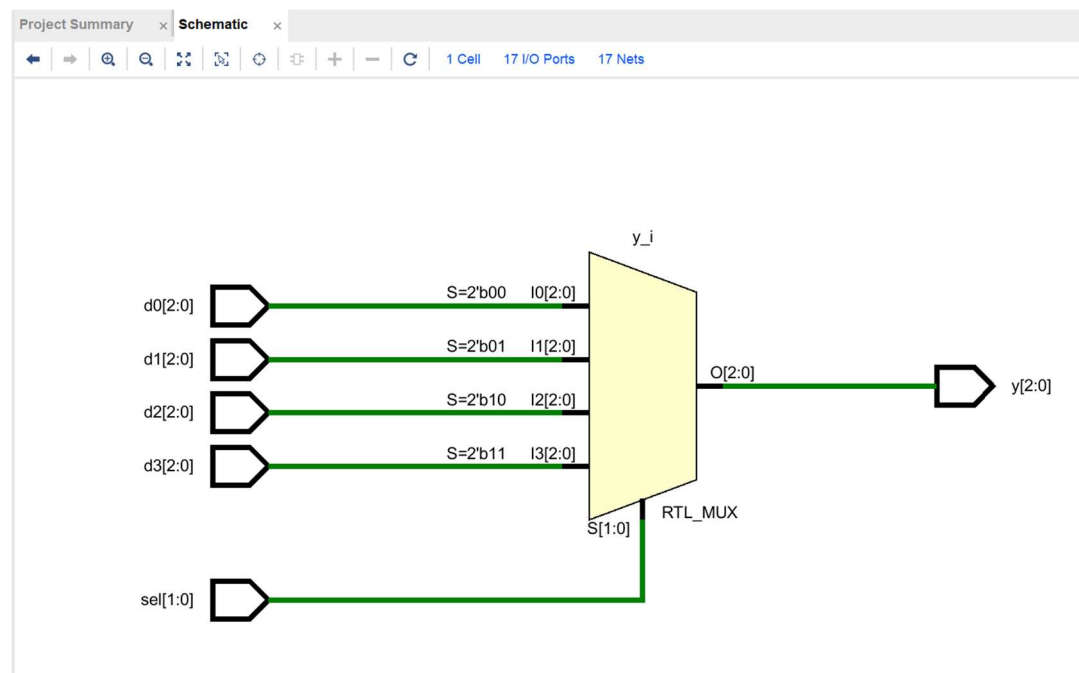
三、实验设计

实验原理：通过 2 位选择控制信号产生 4 种唯一组合，用这些组合作为“选通信号”，其本质是组合逻辑电路中“最小项选通 + 或运算”从 4 路并行输入信号中，根据 2 位选择控制信号的取值，选择其中 1 路信号输出

原理图：



RTL 分析图：



引脚分配:

```
1 | set_property IOSTANDARD LVCMOS33 [get_ports {d0[2]}]
2 | set_property IOSTANDARD LVCMOS33 [get_ports {d0[1]}]
3 | set_property IOSTANDARD LVCMOS33 [get_ports {d0[0]}]
4 | set_property IOSTANDARD LVCMOS33 [get_ports {d1[2]}]
5 | set_property IOSTANDARD LVCMOS33 [get_ports {d1[1]}]
6 | set_property IOSTANDARD LVCMOS33 [get_ports {d1[0]}]
7 | set_property IOSTANDARD LVCMOS33 [get_ports {d2[2]}]
8 | set_property IOSTANDARD LVCMOS33 [get_ports {d2[1]}]
9 | set_property IOSTANDARD LVCMOS33 [get_ports {d2[0]}]
10 | set_property IOSTANDARD LVCMOS33 [get_ports {d3[2]}]
11 | set_property IOSTANDARD LVCMOS33 [get_ports {d3[1]}]
12 | set_property IOSTANDARD LVCMOS33 [get_ports {d3[0]}]
13 | set_property IOSTANDARD LVCMOS33 [get_ports {sel[1]}]
14 | set_property IOSTANDARD LVCMOS33 [get_ports {sel[0]}]
15 | set_property IOSTANDARD LVCMOS33 [get_ports {y[2]}]
16 | set_property IOSTANDARD LVCMOS33 [get_ports {y[1]}]
17 | set_property IOSTANDARD LVCMOS33 [get_ports {y[0]}]
18 | set_property PACKAGE_PIN V16 [get_ports {d0[1]}]
19 | set_property PACKAGE_PIN W16 [get_ports {d0[2]}]
20 | set_property PACKAGE_PIN V17 [get_ports {d0[0]}]
21 | set_property PACKAGE_PIN V15 [get_ports {d1[2]}]
22 | set_property PACKAGE_PIN W15 [get_ports {d1[1]}]
23 | set_property PACKAGE_PIN W17 [get_ports {d1[0]}]
24 | set_property PACKAGE_PIN V2 [get_ports {d2[2]}]
25 | set_property PACKAGE_PIN W13 [get_ports {d2[1]}]
26 | set_property PACKAGE_PIN W14 [get_ports {d2[0]}]
27 | set_property PACKAGE_PIN R3 [get_ports {d3[2]}]
28 | set_property PACKAGE_PIN T2 [get_ports {d3[1]}]
29 | set_property PACKAGE_PIN T3 [get_ports {d3[0]}]
30 | set_property PACKAGE_PIN U1 [get_ports {sel[1]}]
31 | set_property PACKAGE_PIN W2 [get_ports {sel[0]}]
32 | set_property PACKAGE_PIN U19 [get_ports {y[2]}]
33 | set_property PACKAGE_PIN E19 [get_ports {y[1]}]
34 | set_property PACKAGE_PIN U16 [get_ports {y[0]}]
```

四、实验过程或算法

实验步骤：

逻辑门搭建→搭建与门，或门，非门→进行接线→进行烧录→进行测试

核心代码：

```
1 // mux4x3.v
2 module mux4x3 (
3     input [2:0] d0, // 输入数据0
4     input [2:0] d1, // 输入数据1
5     input [2:0] d2, // 输入数据2
6     input [2:0] d3, // 输入数据3
7     input [1:0] sel, // 选择信号
8     output reg [2:0] y // 输出数据
9 );
10
11     always @(*) begin // 组合逻辑过程块，使用自动敏感列表
12         case (sel) // 根据 2-bit 选择信号 sel 选择分支
13             2'b00: y = d0; // sel==00 时，输出 y = d0
14             2'b01: y = d1; // sel==01 时，输出 y = d1
15             2'b10: y = d2; // sel==10 时，输出 y = d2
16             2'b11: y = d3; // sel==11 时，输出 y = d3
17             default: y = 3'b000; // 未匹配时的兜底赋值，避免综合出锁存器
18         endcase // 结束 case
19     end // 结束 always
20
21 endmodule
```

五、实验过程中遇到的问题及解决情况

一：开发板下载后输出与预期不符。

排查：选择信号与开发板按键引脚映射错误（电平逻辑颠倒）。

解决：修改约束文件，将按键输入取反后接入。

二：用 Verilog 结构描述编写 1 位 4 选 1 多路选择器代码时，Vivado 提示“存在未使用的输入端口 d3，逻辑冗余”，但代码中明确包含 d3 的与门逻辑。

排查：检查代码中 d3 的连接，发现 d3 未接入最终的或门导致 d3 信号未参与输出逻辑，被工具判定为冗余。

解决：修改或门实例化语句，重新综合后冗余错误消失，仿真验证 d3 正常输出。

三：用 Verilog 行为描述（case 语句）设计 mux4x3 时，综合工具提示“检测到不必要的锁存器，可能导致时序不稳定”，虽代码能通过综

合，但存在潜在硬件风险。

排查：检查代码中的 case 语句，发现未添加 default 分支，当选择信号出现异常时，输出无法确定，触发锁存器生成。

解决：在 case 语句末尾添加 default 分支，定义异常情况下的输出重新综合后锁存器警告消除，RTL 电路图显示为纯组合逻辑，无锁存器模块，问题解决。

六、实验结果及分析和（或）源程序调试过程

实验结果分析（两种方法对比）：

Block Design 方法

RTL 电路图：由基础逻辑门（与门、或门、非门）组合而成，结构直观，符合电路原理。

仿真波形：输入信号变化时，输出 y 能准确跟随选中的输入信号。

源代码方法（Verilog 结构描述）

RTL 电路图：综合后与 Block Design 结构相似，但布局更紧凑，工具自动优化了冗余逻辑。

仿真波形：与 Block Design 结果一致，验证了逻辑正确性。

七、小组分工情况说明

组员汤英琦：进行 IP 核的设计与封装，进行仿真验证实验的正确性，使用 Verilog HDL 语言的行为描述方法设计一个 3 位数据的 4 选 1 多路选择器 mux4x3。

组员石航恺：使用 Block Design 设计方法和添加源代码的方法设计 4 选 1 多路选择器，并通过仿真、看 RTL 电路图、下载到开发板验证其正确性，使用 Verilog HDL 语言的行为描述方法设计一个 3 位数据的 4 选 1 多路选择器 mux4x3。