

四川大学计算机学院、软件学院

实验报告

学号：2023141460321 姓名：孙谦昊 专业：计算机科学与技术班级：行政七班 第 12 周

课程名称	计算机组成原理实验	实验课时	1-4 节
实验项目	多路数据选择器的设计与实现	实验时间	2024. 11. 21
实验目的	1. 掌握ISE软件设计的方法 2. 掌握数据选择器的工作原理和逻辑功能 3. 掌握简单的Verilog代码编写 4. 掌握schematic使用方法		
实验环境	SWORD 4.0 套件、ISE Design Suite 14.7 、Vivado Design Suite 2014.3 及以上版本		
实 验 内 容 (算法、程序、步骤和方法)	<p>1. 建立新的工程</p> <p>双击桌面上“Xilinx ISE 14.7”图标，启动 ISE 软件，选择 File>New Project 选项，弹出新建工程引导界面。在对话框中输入工程名称 adder_4bit，并指定工程路径。点击 Next 按钮进入下一页选 Kintex7 XC7K325T 芯片，采用 FFG676 封装。另外，选择 Verilog 作为默认的硬件描述语言。再点击 Next 按钮进入下一页确认新建工程的信息。若无误，点击 Finish 。</p> <p>2. 一位 2 选 1 多路选择器设计</p> <p>在工程管区（左上角）任意位置单击鼠标右键，在弹出的菜单中选择 New Source 命令，弹出新建源代码对话框：Select Source Type。选择 Schematic 进行画图，文件命名为 mux2t1。单击 Next 进入下一步，忽略模块的端口定义，点 next 继续，点击 Finish 完成创建。在空白的画布中绘制如下图所示的电路图：</p> <div data-bbox="487 1473 1093 1736"></div> <p>绘制完成电路后，对这个 1 位 2 选 1 数据选择器进行封装。</p> <p>先点击 保存工程，然后在工程区选中 mux2t1，在 process 中展开 Design Utilities，双击 Create Schematic Symbol，生成 mux2t1 的符号 mux2t1. sym</p>		

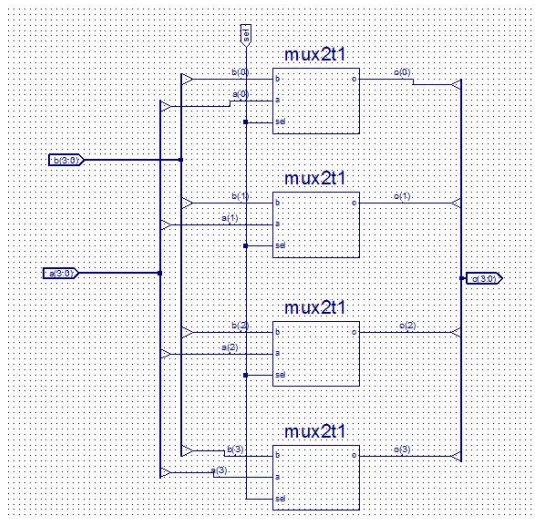
在工程管理区的 View 中选择 Implementation，并选中要综合的模块 mux2t1, 然后在过程管理区中双击 Synthesize-XST，开始综合过程。综合过程结束后，建立仿真文件在工程管理区将 View 设置为 Simulation，在任意位置单击鼠标右键，并在弹出的菜单中选择 New Source。选择要进行测试的模块：mux2t1 模块。点击 Next，单击 Finish 按钮。在弹出的代码页面中，将`ifdef auto_init 和`endif 及其之间的代码都注释掉，替换为：

```
initial begin
a = 0;
b = 0;
sel = 0;
#100;
a = 1;
b = 0;
sel = 0;
#100;
a = 0;
b = 1;
sel = 1;
end
```

完成测试文件编辑并保存后，确认工程管理区中 View 选项设置为 Simulation, 并选中 mux2t1_tb 模块。右键单击其中的 Simulate Behavioral Model 项，选择弹出菜单中的 Process Properties 项，使用默认配置即可。在工程管理区选中测试代码（顶层代码），然后在过程管理区双击 Simulate Behavioral Model 进行仿真。仿真结果与理论值想同后即可进行下一步。以后的步骤均按照此节中的步骤进行。

3. 四位 2 选 1 多路选择器的设计

创建新的文件，命名为 mux2t1_4。电路图设计为：



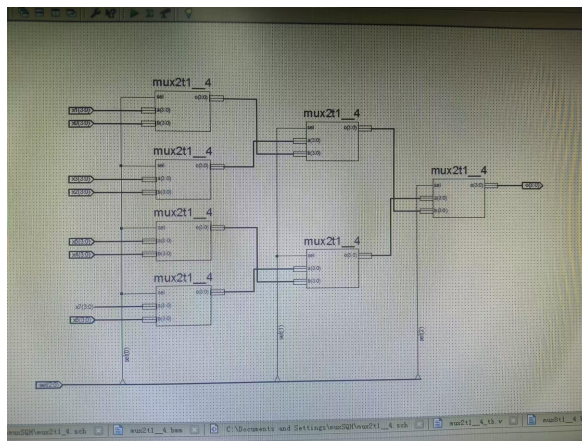
随后进行封装。将测试代码中的`ifdef auto_init 和`endif 及其之间的代码都注释掉，替换为：

```
initial begin
a = 0;
b = 0;
sel = 0;
#100;
a = 4'b1010;
b = 4'b0001;
sel = 0;
#100;
sel = 1;
end
```

随后进行仿真验证，正确进入下一步。

4. 四位 8 选 1 多路选择器的设计

建立 mux8t1_4 文件，绘制如下所示的电路图：



随后进行封装。将测试代码中的`ifdef auto_init 和`endif 及其之间的代码都注释掉，替换为：

```
initial begin
x0 = 4'h0;
x1 = 4'h1;
x2 = 4'h2;
x3 = 4'h3;
x4 = 4'h4;
x5 = 4'h5;
x6 = 4'h6;
x7 = 4'h7;
sel = 0;
```

```

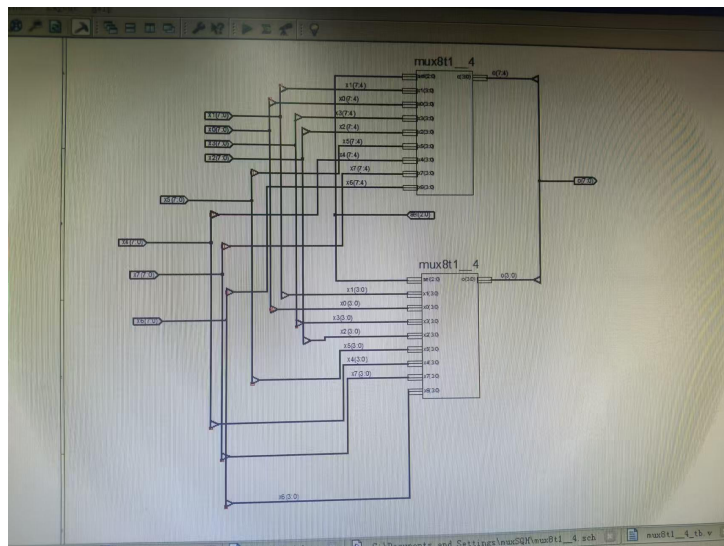
#100;
sel = 1;
#100;
sel = 2;
#100;
sel = 3;
      #100;
sel = 4;
#100;
sel = 5;
#100;
sel = 6;
#100;
sel = 7;
end

```

随后进行仿真验证，正确进入下一步。

5. 八位 8 选 1 多路选择器的设计

建立文件 mux8t1_8，绘制如下所示的电路图：



随后进行封装。将测试代码中的`ifdef auto_init 和`endif 及其之间的代码都注释掉，替换为：

```

initial begin
sel = 0;
x0 = 8'h00;
x1 = 8'h11;
x2 = 8'h22;
x3 = 8'h33;
x4 = 8'h44;

```

```

x5 = 8'h55;
x6 = 8'h66;
x7 = 8'h77;
#100;
sel = 1;
#100;
sel = 2;
#100;
sel = 3;
#100;
sel = 4;
#100;
sel = 5;
#100;
sel = 6;
#100;
sel = 7;
end

```

6. 顶层模块设计

顶层模块的核心是 8 位 8 选 1 多路选择器 mux8t1_8，数据来源于 rom。一个 64 位的数据，可以分为 8 个 8 位宽的数据，作为多路选择器的输入。多路选择器的 8 位输出连接到 8 个 LED，多路选择器的控制信号 sel 为 3 位，来自于拨位开关 sw 的低三位 (2:0)。rom 的地址线输入为 addr，即 sw[3]。图中还有一个 8 位 buf。

rom.v 的代码如下：

```

module rom(
    input addr,
    output [63:0] dout
);
    reg [63:0] data[1:0];
    initial begin
        data[0] = 64'hF8829299_B0A4F9C0;
        data[1] = 64'h8E86A1C6_83889080;
    end
    assign dout = data[addr];
endmodule

```

buf8 的代码如下：

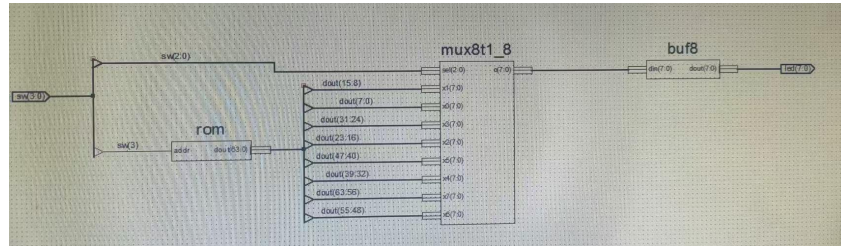
```

module buf8(
    input [7:0] din,
    output [7:0] dout

```

```
);
    assign dout = din;
endmodule
```

绘制如下所示的电路图：

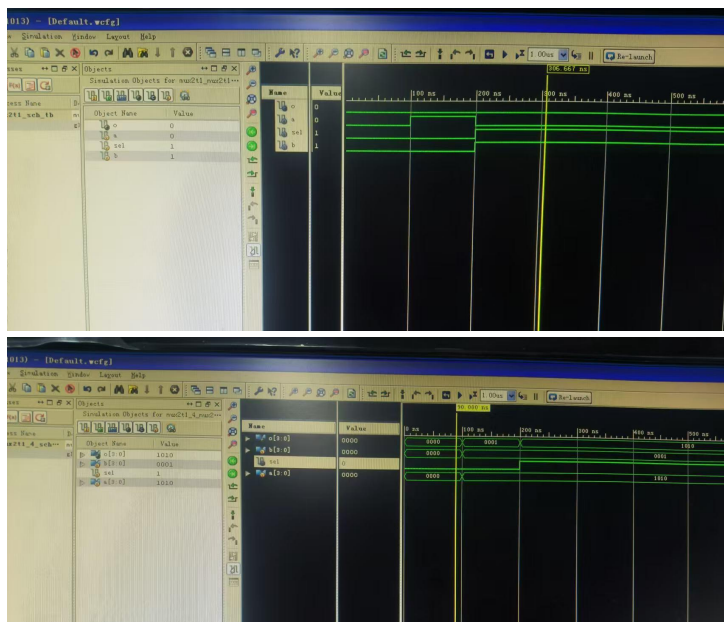


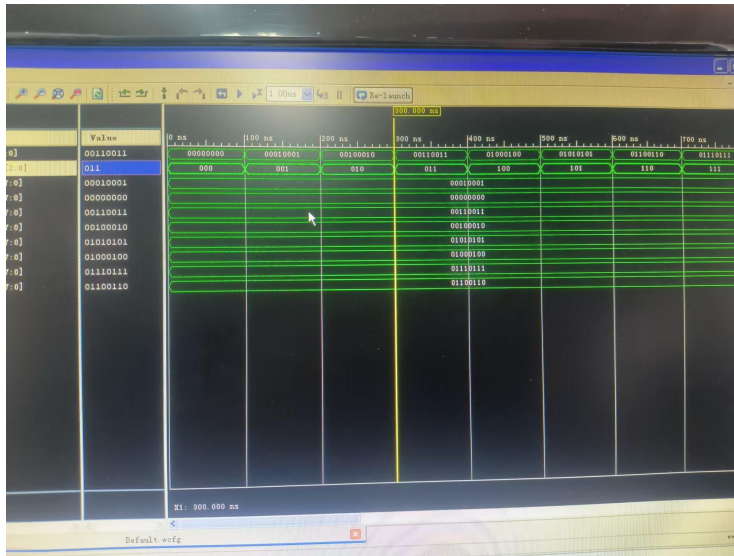
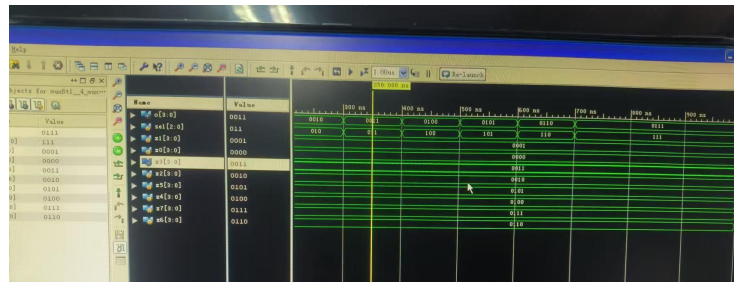
注意：画图连接后，双击下图 2.31 中框线中的总线，将其 Name 改为：dout（63:0）。设计完成后，对该模块进行综合、仿真，建立仿真文件 top_mux_tb.v，测试激励代码如下：

```
initial begin
    sw = 0;
#100;
    sw = 8'h01;
#100;
    sw = 8'h02;
#100;
    sw = 8'h03;
#100;
    sw = 8'h04;
#100;
    sw = 8'h05;
#100;
    sw = 8'h06;
#100;
    sw = 8'h07;
#100;
    sw = 8'h08;
#100;
    sw = 8'h09;
#100;
    sw = 8'h0a;
#100;
    sw = 8'h0b;
#100;
    sw = 8'h0c;
#100;
```

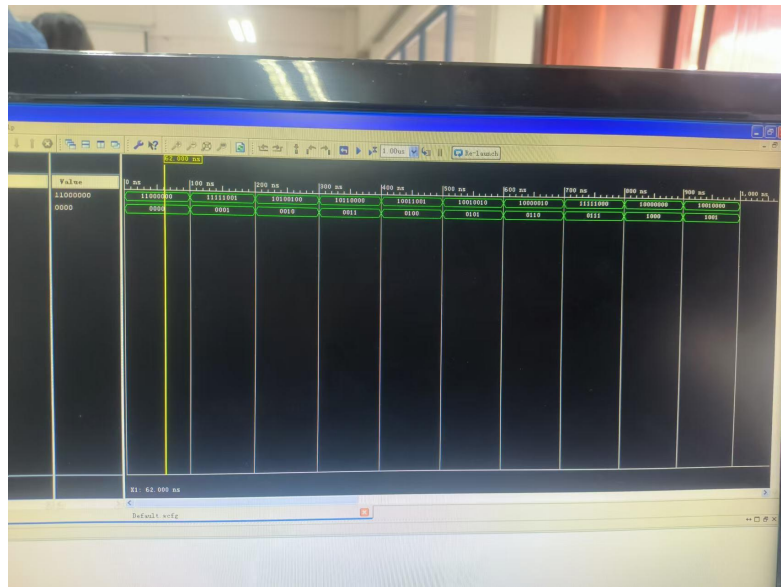
随后进行仿真实验得到对应结果。

数据记录 和计算





最终实验结果为：



结 论 (结 果)	实验结果与理论结果保持一致。
小 结	<p>1.在本实验中，掌握了数据选择器的工作原理和逻辑功能。掌握了简单的 Verilog 代码编写。掌握了 schematic 使用方法。</p> <p>2.复习了多路数据选择器的逻辑电路图和工作原理。</p>
指导老师 评 议	<p>成绩评定：</p> <p>指导教师签名：</p>