

实验二——

姓名：张云策 学号：3200105787 学院：求是学院云峰学园

课程名称：计算机系统 1 同组学生姓名：/

实验时间：2021.4.1 实验地点：紫金港东 4-509 指导老师：吴磊

一、实验目的和要求

- 掌握多路选择器的电路级别逻辑
- 掌握使用 verilog 语言设计多路选择器
- 掌握 testbench 的编写，测试

二、实验内容和原理

2.1 实验内容

1. 使用 Logisim 完成电路级别的多路选择器
2. 使用 Vivado 实现多路选择器

2.2 设计模块

实验一：

1. 多路选择器是数据选择器的别称。在多路数据传送过程中，能够根据需要将其中的任意一路选出来的电路，叫做数据选择器，也称多路选择器或多路开关。
2. 本次实验要求通过最基本的门电路实现已给出文件 multi.circ 中的封装元件原理图，利用选择器性质以及给出真值表，设计出基本门电路。

实验二：

1. 以原有 2 路 32 位选择器为基础，利用 vivado 设计 8 路 32 位选择器。

三、 主要仪器设备

- IDE: Logisim
- IDE: Vivado

四、 操作方法与实验步骤

4.1 操作方法

实验一：

1. 通过设置四个输入值以及 `sw` 值，分别输入不同的 `output`。

实验二：

1. 观察波形图，确认不同输入值可以输出不同的值。

4.2 实验步骤

实验一：

1. 打开 Logisim 软件，打开输入模板文件 `multi.circ`
2. 文件的 `Standard sample` 使用了 1 位的四路选择器，实现了通过改变 `SW` 对输入的选择。
3. 在本次实验中，为了理解多路选择器的电路级别具体实现，放弃使用 Logisim 封装的多路选择器，只采用门级电路，完成 `Standard sample` 的功能实现。

实验二：

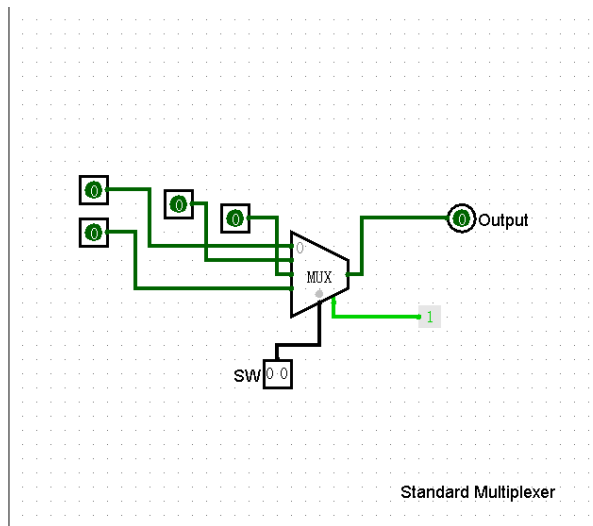
1. 打开 Vivado 软件，按照实验 1 的加入文件方式，加入设计文件样例文件 `Multi_2CH32.v`
2. 同样加入测试文件 `Mul_test.v`
3. 这个测试文件是 32 位 2 路选择器的样例文件，使用 `Run Simulation` 可以仿真。通过更改 `SW` 的值，多路选择器可以选择 `data1` 或者 `data0` 进行输出
4. 本次实验要求，参考设计文件样例文件 `Multi_2CH32.v` 的 32 位 2 路选择器，使用 verilog 代码完成 32 位 8 路选择器。命名为 `Multi_8CH32.v`
5. 将 `SW` 设置成八个输入值，同时将输出值设置成八个不同值，以便区分。

设置 `SW`： `[3:0]SW`

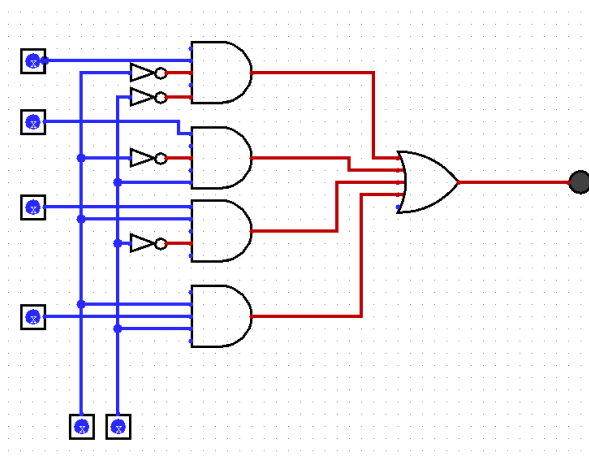
设置 `SW` 区不同值时，输出不同值。

五、 实验结果与分析

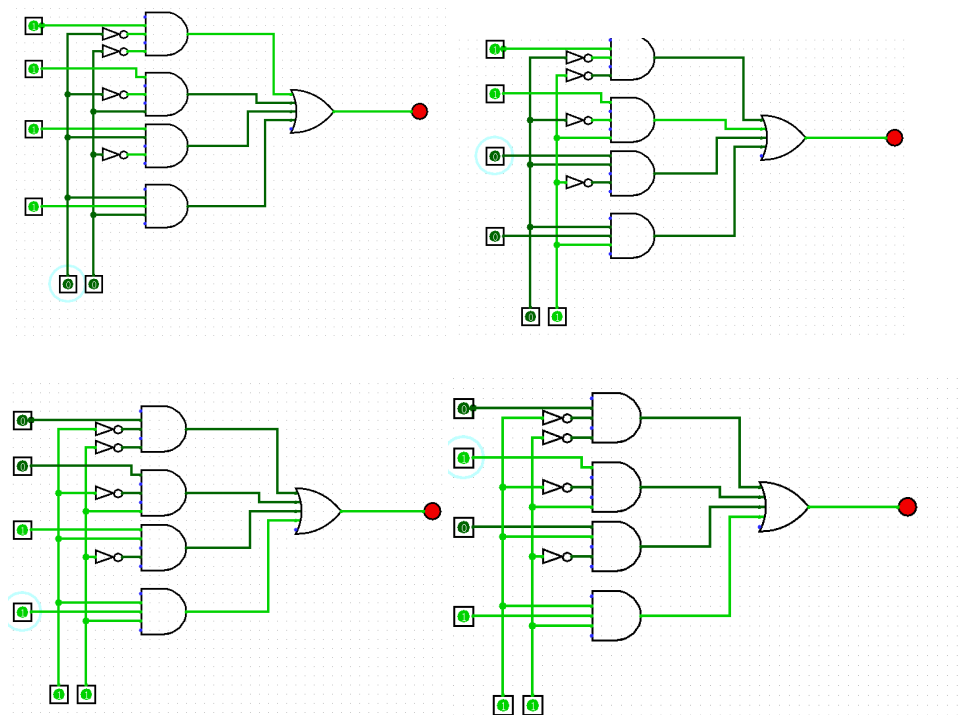
实验一：



为原图



为未封装器件的基本门图。



以上为部分示意图

实验二:

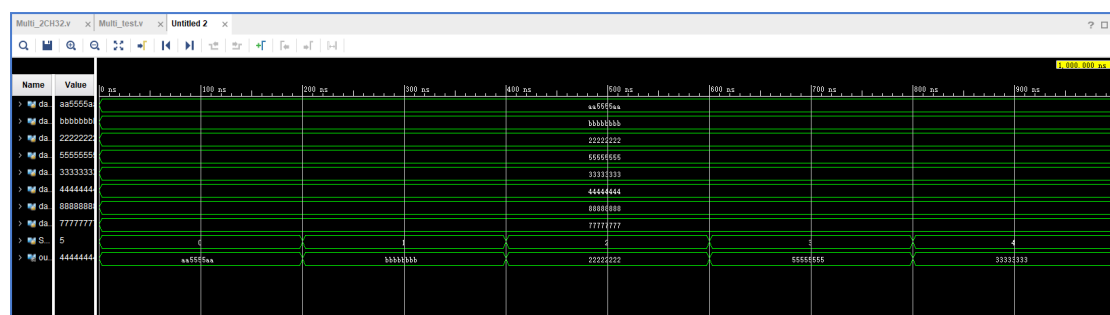
1. 实验代码图: Multi_8CH32

```
module Multi_8CH32(  
    input [3:0] SW, //SW[7:5], 片选信号  
    input [31:0] Data0, //data0-data7均为输入信号  
    input [31:0] Data1,  
    input [31:0] Data2,  
    input [31:0] Data3,  
    input [31:0] Data4,  
    input [31:0] Data5,  
    input [31:0] Data6,  
    input [31:0] Data7,  
  
    output reg [31:0] Disp_num //该信号为输出信号  
);  
    reg [31:0] num_reg = 32'h0;  
    always @ (*)  
    begin  
        case (SW)  
            3'b000: begin Disp_num = Data0; end  
            3'b001: begin Disp_num = Data1; end  
            3'b010: begin Disp_num = Data2; end  
            3'b011: begin Disp_num = Data3; end  
            3'b100: begin Disp_num = Data4; end  
            3'b101: begin Disp_num = Data5; end  
            3'b110: begin Disp_num = Data6; end  
            3'b111: begin Disp_num = Data7; end  
  
        endcase  
    end  
endmodule
```

test

```
module Multi_test(  
    );  
    reg [31:0] data0 = 32'hA55555AA;  
    reg [31:0] data1 = 32'hB55555BB;  
    reg [31:0] data2 = 32'h22222222;  
    reg [31:0] data3 = 32'h55555555;  
    reg [31:0] data4 = 32'h33333333;  
    reg [31:0] data5 = 32'h44444444;  
    reg [31:0] data6 = 32'h88888888;  
    reg [31:0] data7 = 32'h77777777;  
    reg [3:0] SW;  
    wire [31:0] out;  
    Multi_8CH32 U(  
        SW(SW),  
        Data0(data0),  
        Data1(data1),  
        Data2(data2),  
        Data3(data3),  
        Data4(data4),  
        Data5(data5),  
        Data6(data6),  
        Data7(data7),  
        Disp_num(out));  
    initial  
    begin  
        SW = 3'd0;  
        #200  
        SW = 3'd1;  
        #200  
        SW = 3'd2;  
        #200  
        SW = 3'd3;  
        #200  
        SW = 3'd4;  
        #200  
        SW = 3'd5;  
        #200  
        SW = 3'd6;  
        #200  
        SW = 3'd7;  
    end  
endmodule
```

实验波形图 (部分);



六、 讨论、心得

实不相瞒,这次的实验作业经过了学长的指点,他指点了我关于 test 代码如何取值的部分,以及赠予了我一本关于 Verilog 的书籍,若这种行为涉嫌作弊,我会主动承担责任。另外,我个人非常感谢 TA 的热心帮助和指点,让我少走了很多弯路。