实验一 基本门电路的设计

**一、实验目的**

1. 掌握Verilog语言框架，编程及调试的方法
2. 熟悉Verilog的基本语法
3. 熟悉Vivado开发平台
4. 掌握Logisim的使用

**二、实验内容（同时用Logisim和Vivado实现）**

1. 完成一个2输入门电路模块的设计（Verilog赋值语句和Logisim基础电路）。
2. 完成多个（4个以上）门电路之间的级联，形成一个完整的电路。
3. 在Vivado中完成一个工程的设计、编辑、综合和实现的全过程。
4. 掌握以上电路的程序结构和风格。
5. 观察和分析仿真波形，注重输入输出之间的时序关系。
6. 观察logisim真值表（选logisim可忽略3-5）

**三、实验要求**

1. 画出模块的电路图。
2. 分析电路的仿真波形。
3. 记录设计和调试过程。

**四、实验代码及结果**

**Vivado**

**1.2输入与门**

**（1）设计编辑**

**模块文件 and21.v**

` timescale 1ns / 1ps

module and21(D0, D1, Y); //二输入与门

Input D0; //输入端

D0 input D1; //输入端

D1 output Y; //输出端 Y

wire D0, D1, Y;

//数据流描述方式，连续赋值语句

//右边操作数发生变化时，重新计算结果赋值给左侧变量（wire 类型）

assign Y = D0 & D1;  
endmodule

**仿真文件 and21\_simulation.v**

` timescale 1ns / 1ps

module and21\_simulation();

reg D0, D1;

wire Y;

//initial 语句只执行一次，用于初始化。

initial begin

D0 = 0; D1 = 0; //初始值

#10 D0 = 1; D1 = 0; //测试值

#10 D0 = 0; D1 = 1;

#10 D0 = 1; D1 = 1;

#10 D0 = 0; D1 = 0;

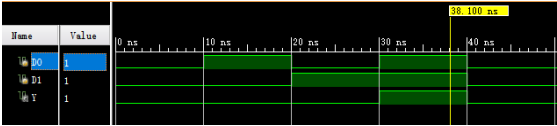
end

//实例化待测模块

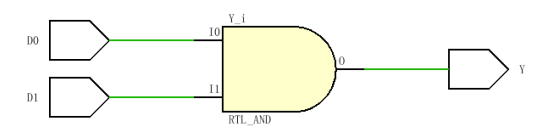
and21 and21t(D0, D1 ,Y);

endmodul

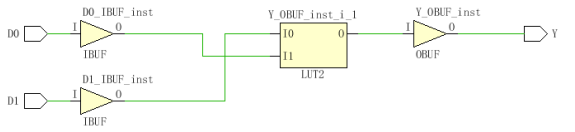
**（2）仿真**



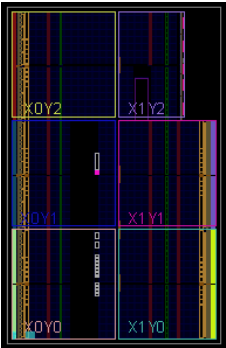
**（3）RTL分析**



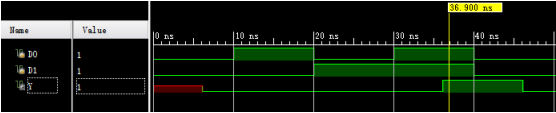
**（4）综合**



**（5）实现**



**（6）实现后时序仿真(Post-Implementation Timing Simulation)**



**2. 6 输入门**

**(1)设计编辑**

**模块文件 and21.v**

` timescale 1ns / 1ps

module and21(D0, D1, Y); //二输入与门

input D0; //输入端 D0

input D1; //输入端 D1

output Y; //输出端 Y

wire D0, D1, Y;

//数据流描述方式，连续赋值语句

//右边操作数发生变化时，重新计算结果赋值给左侧变量（wire 类型）

assign Y = D0 & D1;

endmodul

**模块文件 or21.v**

`timescale 1ns / 1ps

module or21(D0, D1, Y); //二输入或门

input D0; //输入端 D0

input D1; //输入端 D1

output Y; //输出端 Y

wire D0, D1, Y;

//数据流描述方式，连续赋值语句

//右边操作数发生变化时，重新计算结果赋值给左侧变量（wire 类型）

assign Y = D0 | D1;

endmodule

**模块文件 xor21.v**

`timescale 1ns / 1ps

module xor21(D0, D1, Y); //二输入异或门

input D0; //输入端 D0

input D1; //输入端 D1

output Y; //输出端 Y

wire D0, D1, Y;

//数据流描述方式，连续赋值语句

//右边操作数发生变化时，重新计算结果赋值给左侧变量（wire 类型）

assign Y = (~D0 & D1) | (D0 & ~D1);

endmodule

**仿真文件 and21\_simulation.v**

`timescale 1ns / 1ps

module door61\_simulation();

reg D0, D1, D2, D3, D4, D5; //定义测试输入信号

wire Y; //输出

//initial 语句只执行一次，用于仿真时变量初始化，生成激励波形

initial begin //顺序语句块 begin...end，语句在 0 时刻开始，顺序执行

D0=0; D1=0; D2=0; D3=0; D4=0; D5=0; //初始化赋值

#5 {D0, D1, D2, D3, D4, D5}=6'b100000; //位拼接赋值

fork //并行语句块，语句并行执行

repeat(5) #5 D1=~D1;

repeat(5) #10 D2=~D2;

join

end

initial fork //并行语句块，从 0 时刻开始，语句并行执行

repeat(10) #15 D3=~D3;

repeat(10) #20 D4=~D4;

join

always begin //循环执行

#10 D5=~D5;

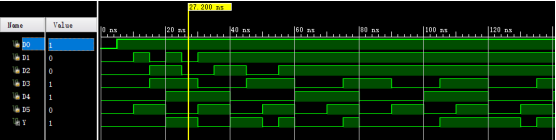
end

//实例化待测模块

door61 door61\_1(D0, D1, D2, D3, D4, D5, Y);

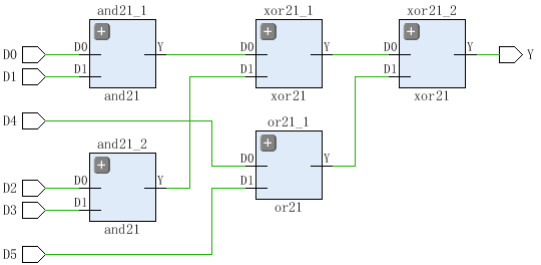
endmodule

**（2）仿真**

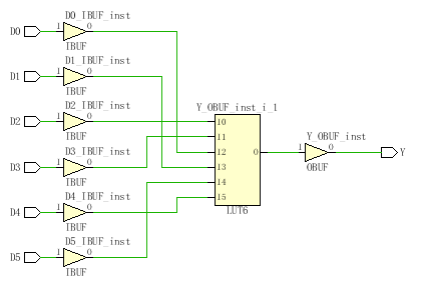


将仿真结果与逻辑运算结果进行对比，验证了设计的正确性。

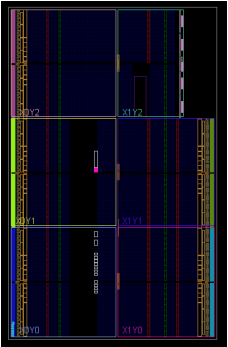
**（3）RTL分析**



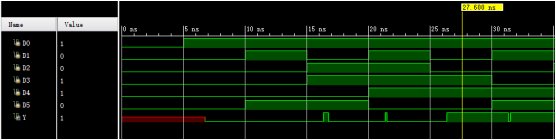
**（4）综合**



**（5）实现**



**（6）实现后时序仿真(Post-Implementation Timing Simulation)**

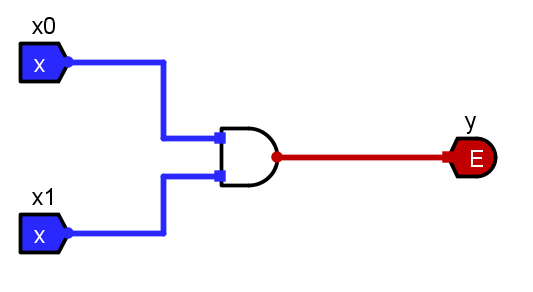


与前仿真不同的是，实现后时序仿真模拟了真实情况下器件的工作状态，各个门电路都存在一定的时延，导致输出信号出现毛刺和错误。

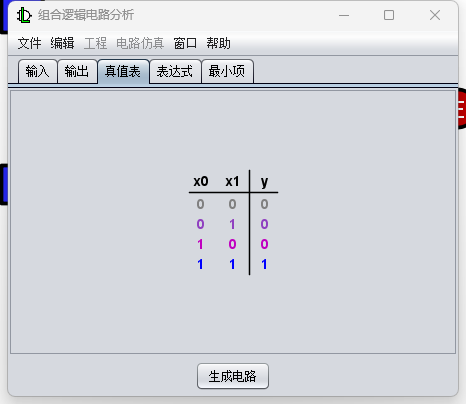
**Logisim**

1. **完成一个2输入门电路模块的设计**

**（1）电路图设计**

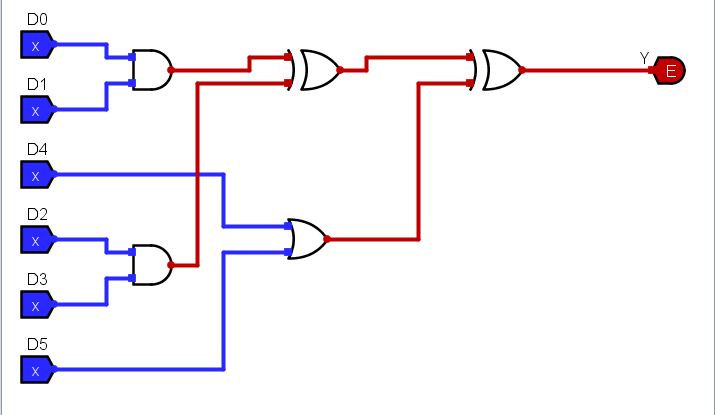


**（2）真值表分析**



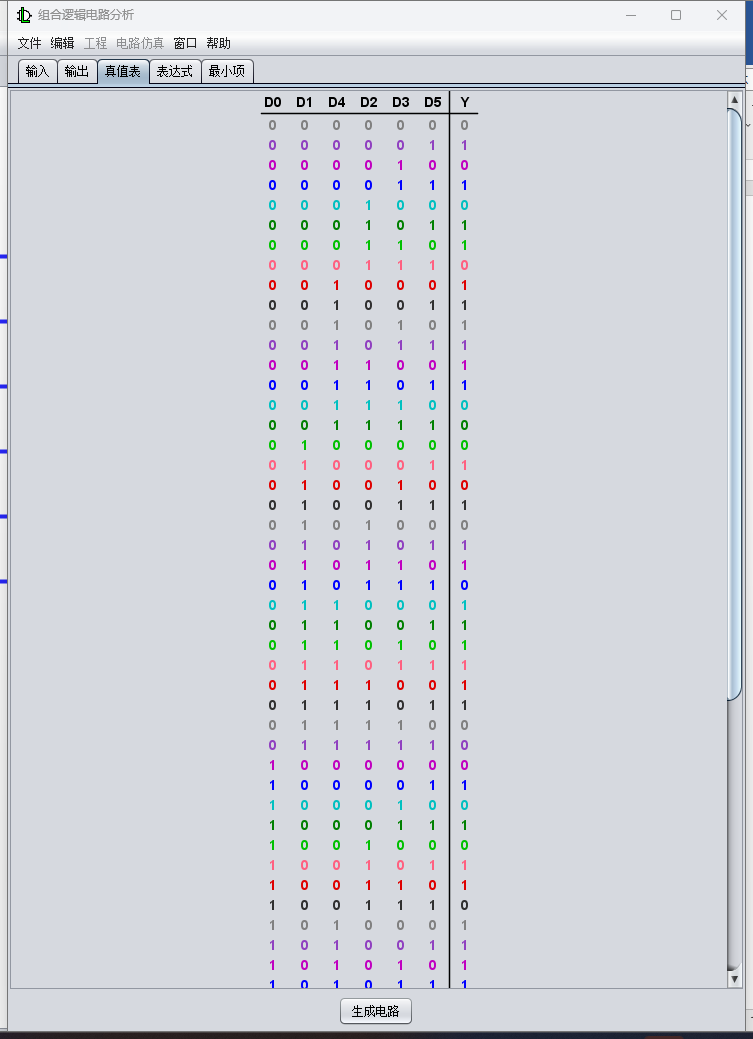
可以看出符合要求

1. **完成多个（4个以上）门电路之间的级联，形成一个完整的电路**
2. **电路图设计**



1. **真值表分析**





**五、调试和心得体会**

1. 对 Verilog 的基本语句有了一定的了解，

2. 根据老师讲授的内容和视频学习，熟悉了 Vivado 的设计编辑、仿真、RTL 分析、综合、实现以及实现后时序仿真等一系列操作的方法。

3. 回忆起之前学习过的数字电路知识和硬件编程语言。

4. 学会了使用新软件logisim