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. 掌握iverilog或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iverilog或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。

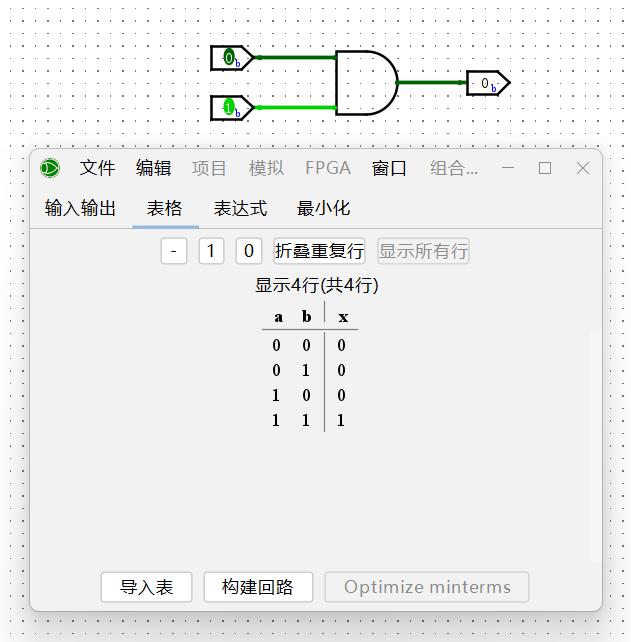
**三、实验要求**

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**

1）二输入与门

Logisim实现（真值表方法）



Vivado实现

代码：

module and21(D0, D1, Y);

input D0;

input D1;

output Y;

wire D0, D1, Y;

assign Y = D0 & D1;

endmodule

module sim\_and21;

reg D0, D1;

wire Y;

initial begin

D0 = 0; D1 = 0;

#10 D0 = 1; D1 = 0;

#10 D0 = 0; D1 = 1;

#10 D0 = 1; D1 = 1;

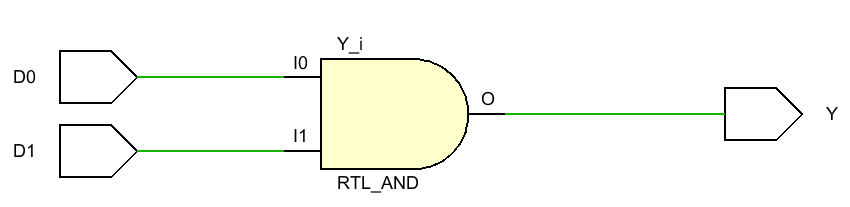
#10 D0 = 0; D1 = 0;

end

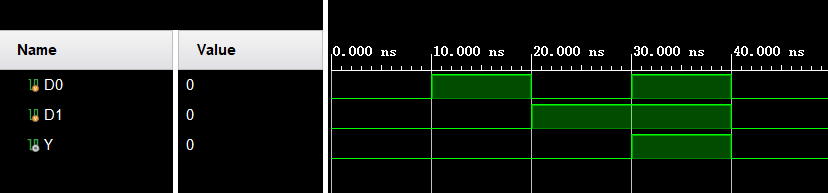
and21 and21t(D0, D1, Y);

endmodule

电路图：

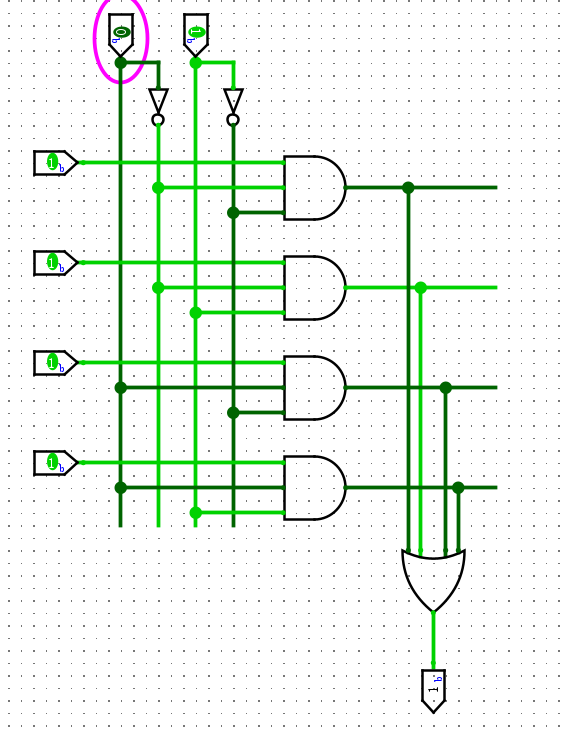


仿真波形：



2）复用器

Logisim实现（电路图方法）



Vivado实现

代码：

module mux41(

input D0, input D1, input D2, input D3,

input [1:0] S,

output Y

);

reg temp;

always@(\*)

begin

case(S)

2'b00: temp = D0;

2'b01: temp = D1;

2'b10: temp = D2;

2'b11: temp = D3;

default: temp = D0;

endcase

end

assign Y = temp;

endmodule

module sim\_mux41;

reg D0, D1, D2, D3;

reg [1:0] S;

wire Y;

mux41 mux(D0, D1, D2, D3, S, Y);

initial

begin

D0 = 0; D1 = 0; D2 = 0; D3 = 0; S = 2'b00;

fork

repeat(100) #10 D0 = ~D0;

repeat(50) #20 D1 = ~D1;

repeat(25) #40 D2 = ~D2;

repeat(10) #100 D3 = ~D3;

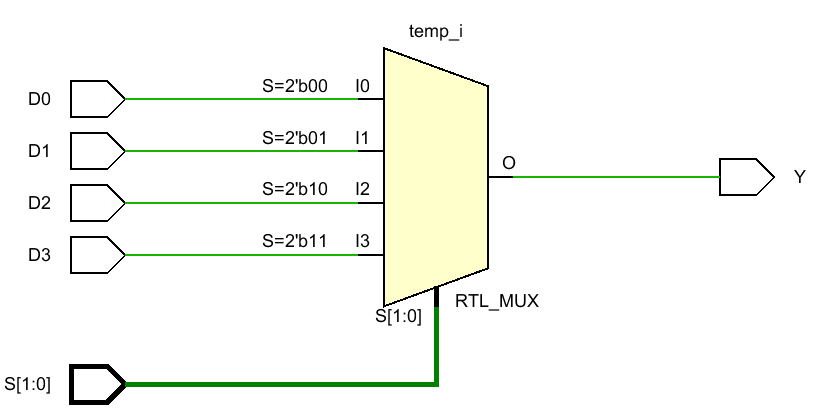
repeat(5) #200 S = S + 1;

join

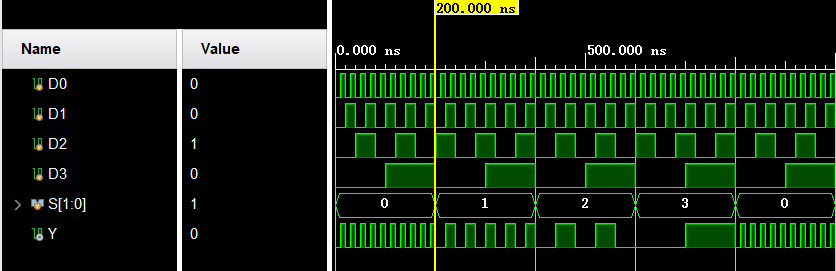
end

endmodule

电路图：

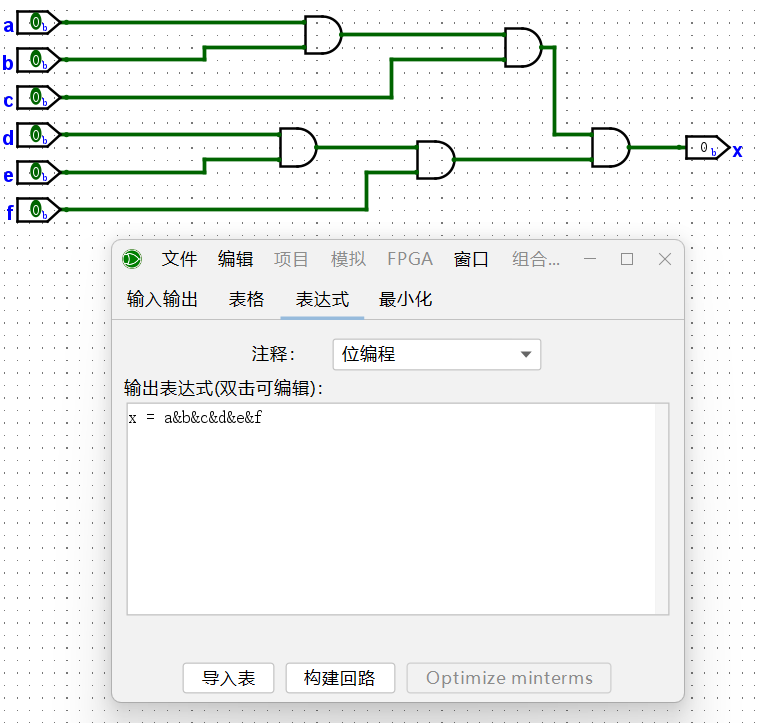


仿真波形：



3）六输入与门

Logisim实现（逻辑表达式方法）



Vivado实现

代码：

module and61(

input D0, D1, D2, D3, D4, D5,

output Y

);

wire Y1, Y2, Y3, Y4;

and21 and21\_1(D0, D1, Y1),

and21\_2(D2, D3, Y2),

and21\_3(D4, D5, Y3),

and21\_4(Y1, Y2, Y4),

and21\_5(Y3, Y4, Y);

endmodule

module sim\_and61();

reg D0, D1, D2, D3, D4, D5;

wire Y;

integer i;

initial begin

D0 = 0; D1 = 0; D2 = 0; D3 = 0; D4 = 0; D5 = 0;

#5 {D0, D1, D2, D3, D4, D5} = 6'b100000;

fork

forever #5 D1 = ~D1;

repeat(5) #10 D2 = ~D2;

join

end

initial fork

while(1) begin #15 D3 = ~D3; end

for(i = 0; i < 10; i = i + 1) #20 D4 = ~D4;

join

always begin

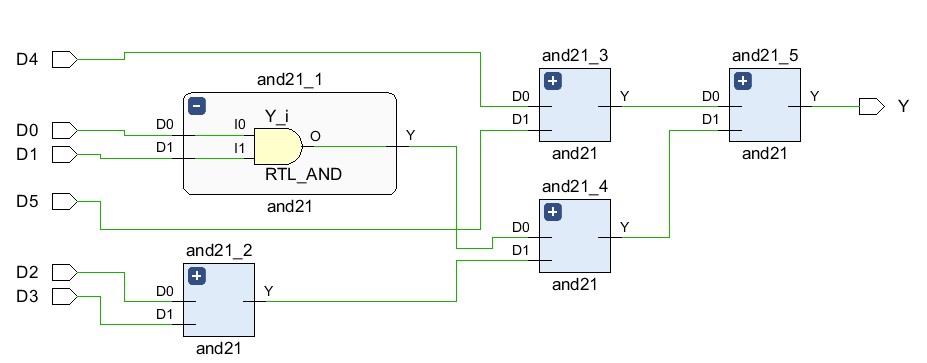
#10 D5 = ~D5;

end

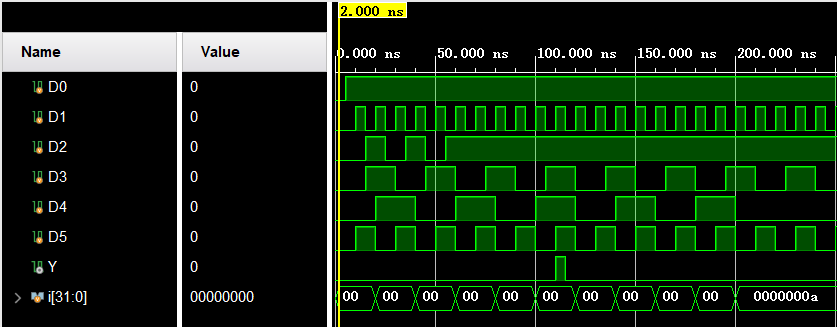
and61 and61t(D0, D1, D2, D3, D4, D5, Y);

endmodule

电路图：



仿真波形：



**五、调试和心得体会**

通过本次实验，我掌握了Vivado的使用方法，特别是verilog语言的编程方式，熟悉了一些基本语句，比如：assign赋值语句、fork并行语句块、forever循环语句等，还了解了如使用Vivado进行波形仿真与电路生成。此外，我还掌握了Logisim的基本使用，能够通过真值表、逻辑表达式、电路图等方式完成数字逻辑电路的设计。