

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）

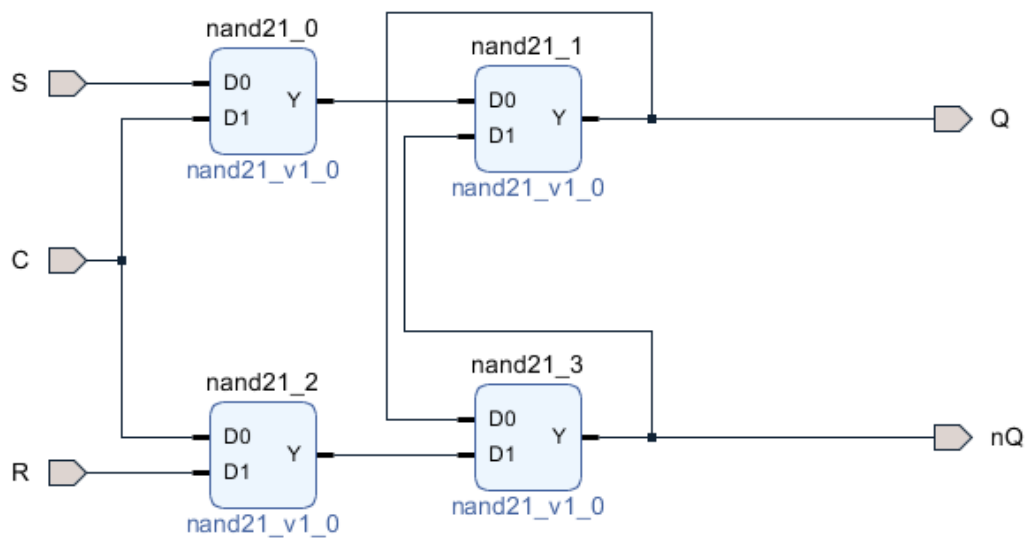
三、实验要求

1. 画出模块的电路图。

2 输入门电路模块

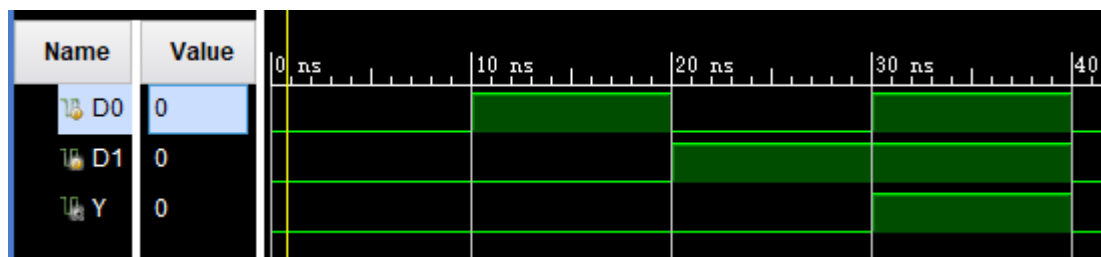


带使能端的 S-R 锁存器



2. 分析电路的仿真波形。

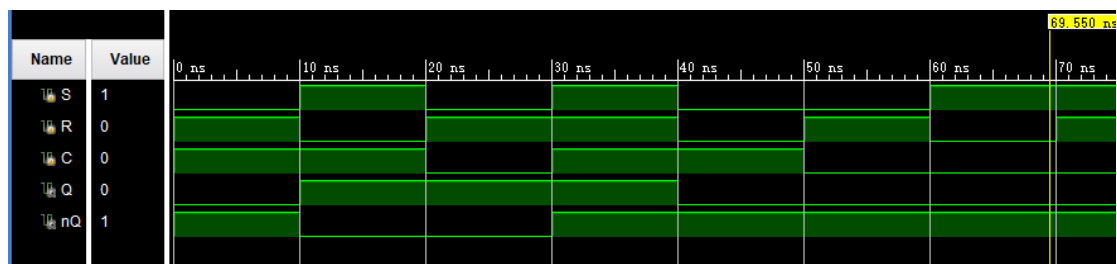
2 输入门电路模块



真值表为

D0	D1	Y
0	0	0
0	1	0
1	0	0
1	1	1

带使能端的 S-R 锁存器



当使能端 C 无效时，为锁存状态，Q 端与 nQ 端保留原值 Q_{pre} 和 nQ_{pre} 不变。

当使能端 C 有效时，且 S 端与 R 端不一致时，Q 端与 S 端状态一致，nQ 与 R 端一致。

当使能端 C 有效，S 端与 R 端都无效时，为锁存状态，Q 端与 nQ 端保留原值 Q_{pre} 和 nQ_{pre} 不变。

当使能端 C 有效，S 端与 R 端都有效时，Q 端与 nQ 端状态不定。

真值表为

R	S	C	Q	nQ
0	0	d	Q_{pre}	nQ_{pre}
d	d	0	Q_{pre}	nQ_{pre}
1	0	1	1	0
0	1	1	0	1
1	1	0	x	x

3. 记录设计和调试过程。

首先设计一个二输入与非门，即电路图中 nand21 模块。将 nand21 模块进行 IP 封装，然后按照电路图进行 IP 设计出带使能端的 S-R 锁存器

四、实验代码及结果

2 输入门电路模块

and21.v

```
module and21(D0, D1, Y); //二输入与门
```

```
    input D0;
```

```
    input D1;
```

```
    output Y;
```

```
    wire D0, D1, Y;
```

```
    assign Y = D0 & D1;
```

```
endmodule
```

```
sim_and21.v
```

```
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
```

```
sr.v
```

```
(*                                CORE_GENERATION_INFO                                =  
"sr, IP_Integrator, {x_ipVendor=xilinx.com, x_ipLibrary=BlockDiagram, x_i  
pName=sr, x_ipVersion=1.00.a, x_ipLanguage=VERILOG, numBlks=4, numReposBl  
ks=4, numNonXlnxBlks=0, numHierBlks=0, maxHierDepth=0, numSysgenBlks=0, nu  
mHlsBlks=0, numHdlrefBlks=0, numPkgbdblks=0, bdsourc=USER, synth_mode=00  
C_per_IP}" *) (* HW_HANDOFF = "sr.hwdef" *)
```

```
module sr
```

```
    (C,
```

```
    Q,
```

```
    R,
```

```
    S,
```

```
    nQ);
```

```
    input C;
```

```
    output Q;
```

```
    input R;
```

```
    input S;
```

```
    output nQ;
```

```

wire C_1;
wire R_1;
wire S_1;
wire nand21_0_Y;
wire nand21_1_Y;
wire nand21_2_Y;
wire nand21_3_Y;

assign C_1 = C;
assign Q = nand21_1_Y;
assign R_1 = R;
assign S_1 = S;
assign nQ = nand21_3_Y;
sr_nand21_0_0 nand21_0
    (.D0(S_1),
     .D1(C_1),
     .Y(nand21_0_Y));
sr_nand21_1_0 nand21_1
    (.D0(nand21_0_Y),
     .D1(nand21_3_Y),
     .Y(nand21_1_Y));
sr_nand21_2_0 nand21_2
    (.D0(C_1),
     .D1(R_1),
     .Y(nand21_2_Y));
sr_nand21_3_0 nand21_3
    (.D0(nand21_1_Y),
     .D1(nand21_2_Y),
     .Y(nand21_3_Y));
Endmodule

```

```

sim_sr.v
module sim_sr;
    reg S, R, C;
    wire Q, nQ;
    initial begin
        {S, R, C} = 3'b011;
        #10 {S, R, C} = 3'b101;
        #10 {S, R, C} = 3'b010;
        #10 {S, R, C} = 3'b111;
        #10 {S, R, C} = 3'b001;
        #10 {S, R, C} = 3'b010;
        #10 {S, R, C} = 3'b100;
    end
endmodule

```

```
        #10 {S, R, C} = 3'b110;  
    end  
    sr srt(C, Q, R, S, nQ);  
endmodule
```

五、调试和心得体会

在本次实验中，复习了上学期所学 vivado 基本操作，并完成了 IP 封装。