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

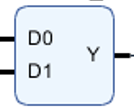
**二、实验内容（同时用Logisim和Vivado实现）**

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）

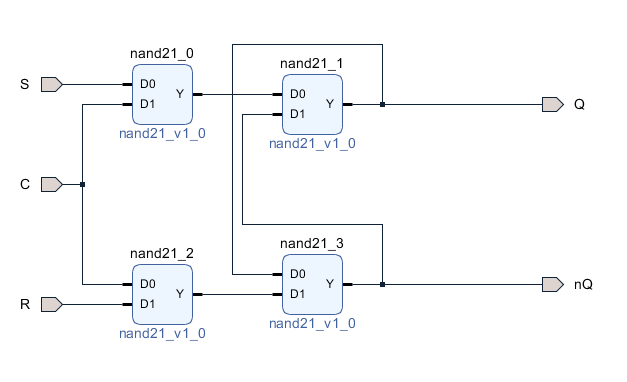
**三、实验要求**

1. 画出模块的电路图。

2输入门电路模块

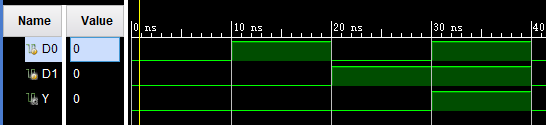


带使能端的S-R锁存器



1. 分析电路的仿真波形。

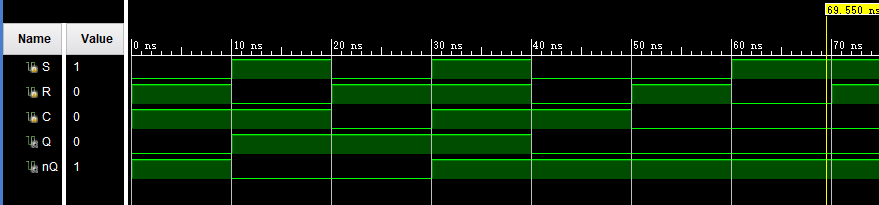
2输入门电路模块



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Qpre和nQpre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Qpre和nQpre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 | Qpre | nQpre |
| d | d | 0 | Qpre | nQpre |
| 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | x | x |

1. 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\_ipName=sr,x\_ipVersion=1.00.a,x\_ipLanguage=VERILOG,numBlks=4,numReposBlks=4,numNonXlnxBlks=0,numHierBlks=0,maxHierDepth=0,numSysgenBlks=0,numHlsBlks=0,numHdlrefBlks=0,numPkgbdBlks=0,bdsource=USER,synth\_mode=OOC\_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;

#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。