

2020-9-20

数字电路实验四 触发器和锁存器

秦嘉余

191220088

1348288404@qq.com

目录

1.实验目的	1
2.实验原理	1
3.实验环境/器材	1
4. 分析阻塞和非阻塞 RTL 视图和仿真结果.....	2
5.设计一个同步清零和一个异步清零的 D 触发器.....	3
1 设计思路	3
2.实验代码	3
3.RTL 视图	4
4.仿真测试代码及仿真图	5
5.引脚分配图	5
6.实验结果	5
6.实验实验中遇到的问题及解决办法.....	6

1.实验目的

本次实验的目的是实现锁存器和触发器这两种时序电路的基本构件。并学习如何对时序电路进行仿真，了解 Verilog 语言中阻塞 赋值语句和非阻塞赋值语句的区别。

2.实验原理

锁存器和触发器分别为电平触发和时钟边沿触发。对于各种锁存器与寄存器，例如 D,JK,RS,T 等等，这些器件仅仅为改变状态的方式不同，但是都符合电平和时钟边沿触发的特性。在 verilog 中，对于 always 语句中，对于的触发条件分别为*与 clock 即可完成触发的条件，这就是利用 verilog 实现锁存器和触发器的实验原理

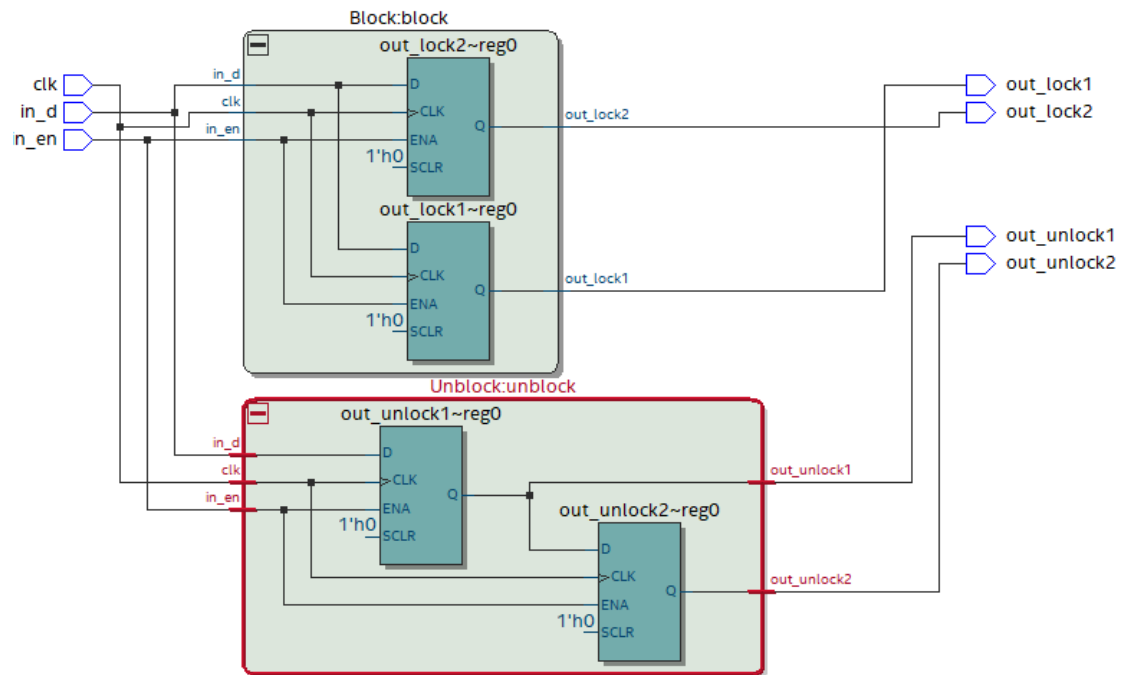
3.实验环境/器材

本次实验的环境为 Quartus17.1 版本

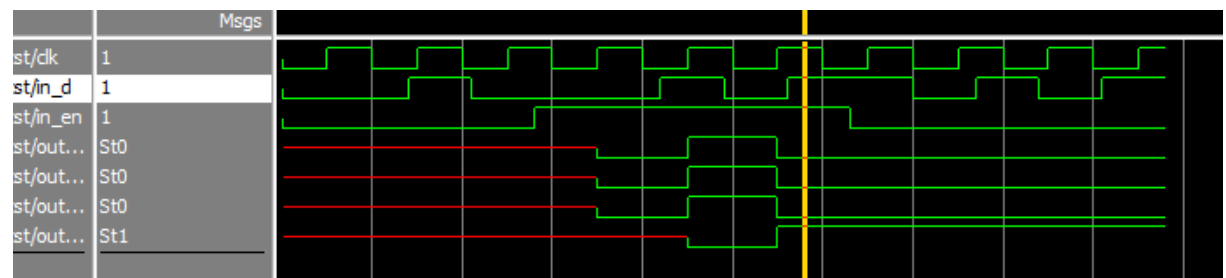
本次实验的器材为 DE10 Standard 开发板

4. 分析阻塞和非阻塞 RTL 视图和仿真结果

文件见压缩包，RTL 视图如下：



上为阻塞赋值，下为非阻塞赋值，模拟仿真如下：



5.设计一个同步清零和一个异步清零的D触发器

1 设计思路

同步清零和异步清零的 D 触发器的区别在于，同步清零的清零端必须在时钟有效的时候才能清零，所以为同步清零，而异步清零在 clr 有效时直接清零，与时钟不同步，所以为异步清零

2.实验代码

顶层模块为 D_trigger，如下：包含两个下层模块，分别为同步清零与异步清零的 D 触发器，参数分别为时钟信号 clk，输入 in_d，使能端 en，清零端 clr，与各自的输出 out_q

```
module D_trigger(clk,in_d,en,clr,out_q1,out_q2);
    input clk;
    input in_d;
    input en;
    input clr;
    output out_q1;
    output out_q2;

    D_synctrigger D_trigger1(.clk(clk),.in_d(in_d),.en(en),.clr(clr),.out_q(out_q1));
    D_asynctrigger D_trigger2(.clk(clk),.in_d(in_d),.en(en),.clr(clr),.out_q(out_q2));
endmodule
```

下图分别为同步清零与异步清零的 D 触发器模块的代码

```
module D_synctrigger(clk,in_d,en,clr,out_q);
    input clk;
    input in_d;
    input en;
    input clr;
    output reg out_q;

    always @(posedge clk )
    begin
        if(en)
        begin
            if(clr)
                out_q<=0;
            else
                out_q<=in_d;
            end
        else
            out_q<=out_q;
        end
    end
endmodule
```

```

module D_asyncntrigger(clk,in_d,en,clr,out_q);
input clk;
input in_d;
input en;
input clr;
output reg out_q;
always @(posedge clk or posedge clr)
begin
if(clr)
begin
if(en)
out_q<=0;
else
out_q<=out_q;
end
else
begin
if(en)
out_q<=in_d;
else
out_q<=out_q;
end
end

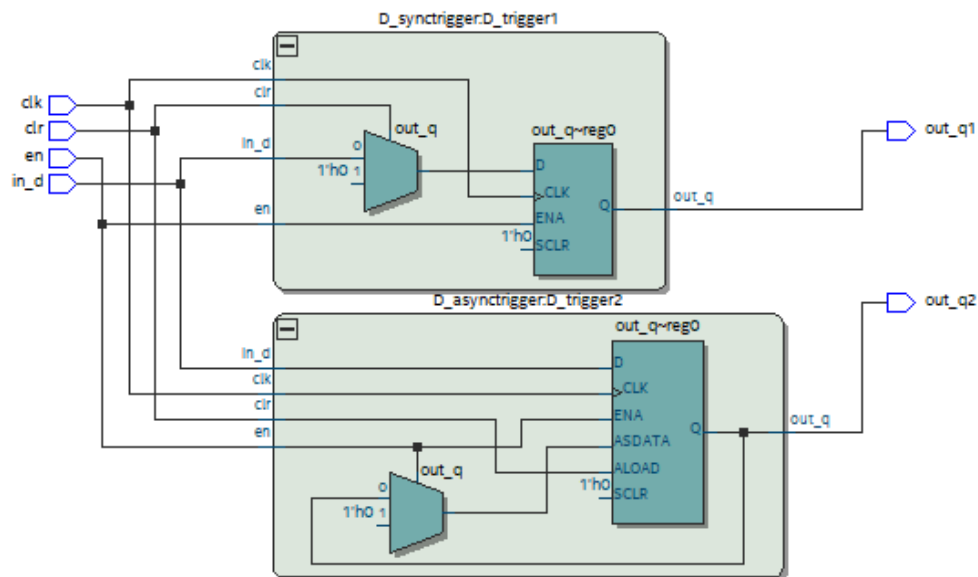
end

endmodule

```

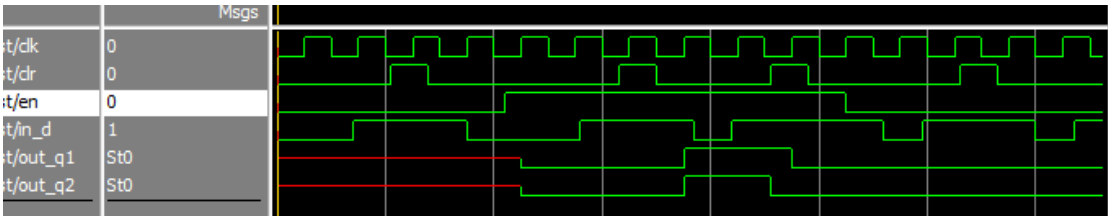
其中最大的区别为 always 语句的触发条件，同步清零为 clr，异步清零则多包含了 clr

3.RTL 视图



4.仿真测试代码及仿真图

```
//initial
begin
// code that executes only once
// insert code here --> begin
  clk=0; in_d=0; en=0; clr=0; #7;
      in_d=0; #7;
      in_d=1; #7;
      clr=1; #7;
      clr=0; #7;
      in_d=0; #7;
  en=1; #7;
      in_d=0; #7;
      in_d=1; #7;
      clr=1; #7;
      clr=0; #7;
      in_d=0; #7;
      in_d=1; #7;
      clr=1; #7;
      clr=0; #7;
  en=0; #7;
      in_d=0; #7;
      in_d=1; #7;
      clr=1; #7;
      clr=0; #7;
      in_d=0; #7;
      in_d=1; #7;
      $stop;
end
always
begin
  #5 clk=~clk;
end
endmodule
```



5.引脚分配图

in	clk	Input	PIN_AJ4	3
in	clr	Input	PIN_AB28	5
in	en	Input	PIN_AB30	5
in	in_d	Input	PIN_Y27	5
out	out_q1	Output	PIN_AA24	5
out	out_q2	Output	PIN_AB23	5

6.实验结果

已在开发板上验收，与仿真结果一致

6. 实验实验中遇到的问题 及解决办法

起初，不理解阻塞赋值与非阻塞赋值的区别，在网上查阅后，明白了其区别，在硬件中阻塞与非阻塞的连接的不同之处，也更加体会到硬件语言与普通编程语言的区别。