**实验四 触发器和锁存器**

**实验报告**

**181860085 汤昊**

**数字电路与数字系统2班**

**邮箱：1174639585@qq.com**

**2019.9.25**

一．实验目的

（1）探究阻塞式和非阻塞式赋值的区别

（2）分别设计两个同步清零触发器和异步清零触发器

二．实验原理

（1）阻塞式赋值和非阻塞式赋值的区别：阻塞式赋值按序进行，非阻塞式赋值并行执行

（2）D触发器特征方程：Q\*=D

三．实验环境及器材

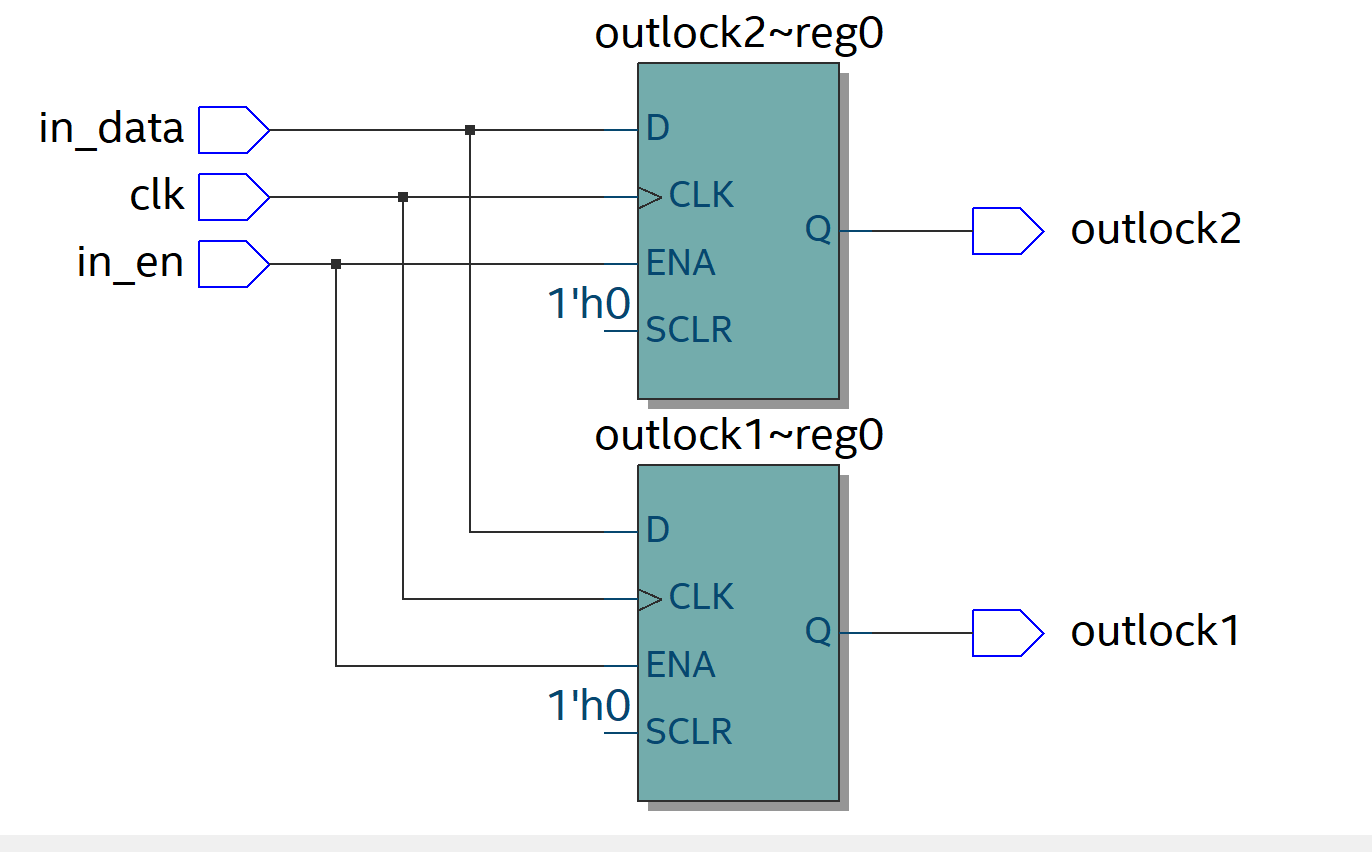
开发软件：quartus prime 17.1

开发器材：DE-standard 开发板

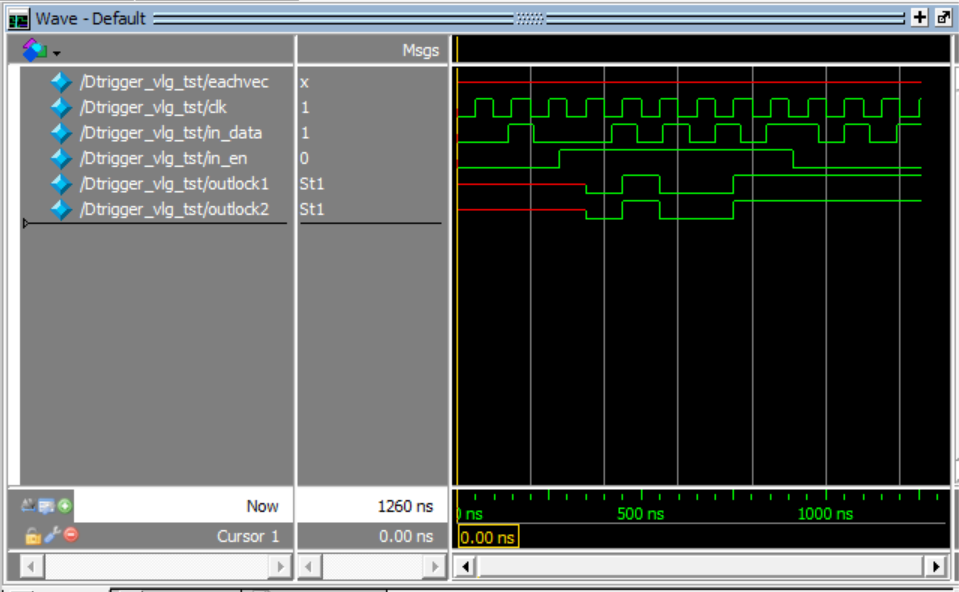
四．阻塞和非阻塞RTL视图和仿真结果

（1）阻塞式

RTL视图：

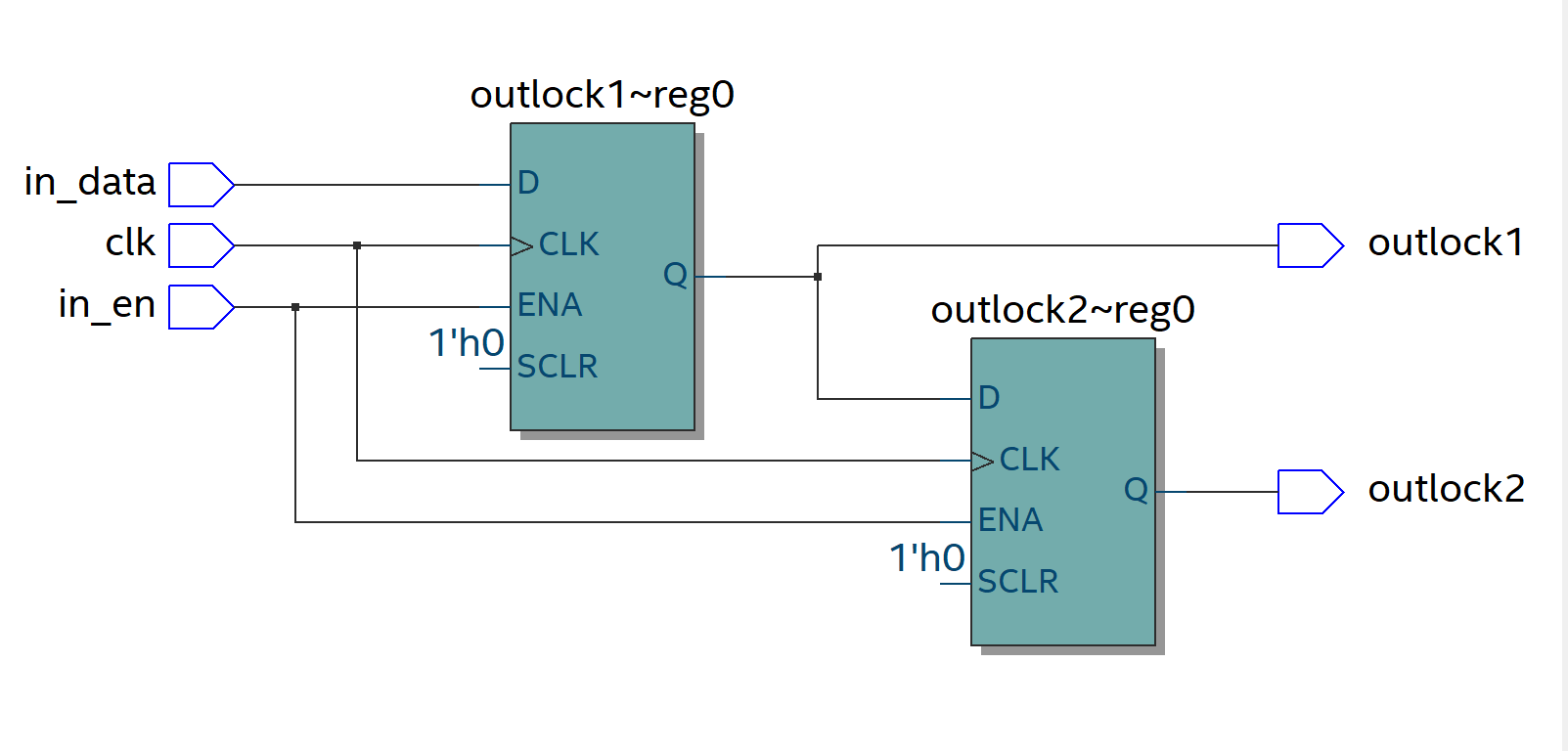


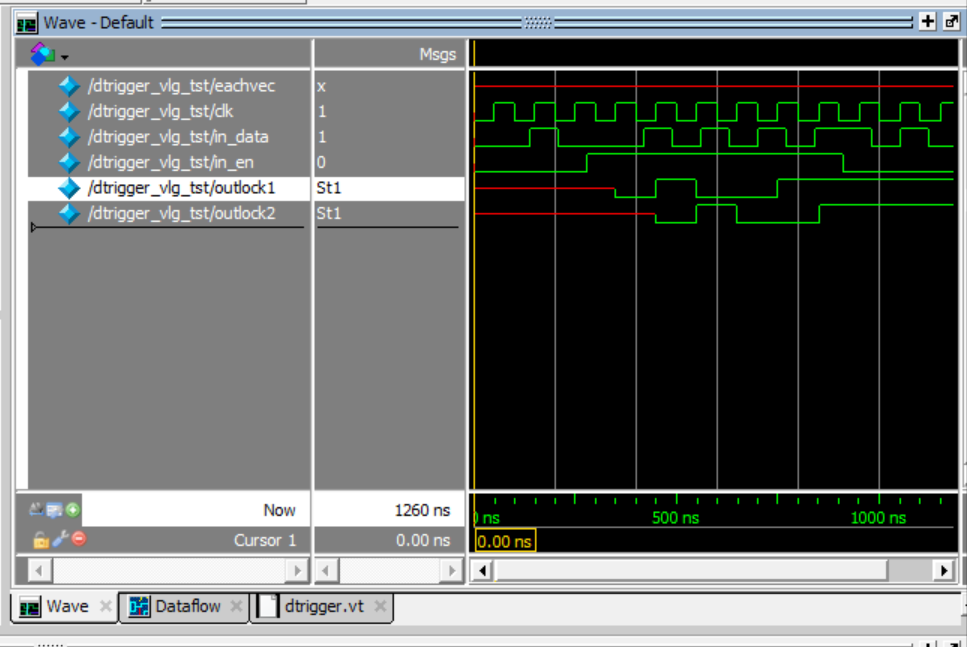
仿真图：



（2）非阻塞式

RTL视图：





在硬件实现电路上的不同：对于阻塞式赋值，两个触发器是并联的，每个触发器有相同的输入，输出也互不干扰，而对于非阻塞式赋值，两个触发器是级联的，触发器1的输出作为触发器2的输入，其余输入相同，这是由于阻塞式赋值是将等号右边的值立即赋给左边，而非阻塞式赋值在计算出右边的结果后，等待always语句结束后才赋给等式左边，因此就有了传导链：数据-第一个触发器-第二个触发器，因此需要用级联的方式来实现这种依赖关系，在仿真模拟的结果上也体现了这种结果的时间差。

五．同步清零触发器和异步清零触发器

同步清零触发器

Trigger(顶层)

异步清零触发器

（1）时钟和清零端设计

Clk上升沿触发，clr\_n低电平有效

（2）同步清零触发器

Clr\_n有效时要等到clk上升沿才会清零，模块命名为mysynchro

* 测试代码

测试方法：通过变换clr\_n和in的输入检测触发器的功能

clk=0;in=0;clr\_n=0;#7;

in=1;#7; //清零端有效，应不做反应

clr\_n=1;#14;//清零端无效

clr\_n=0;#14;//重复测试

in=1;#7;

in=0;#7;

$stop;

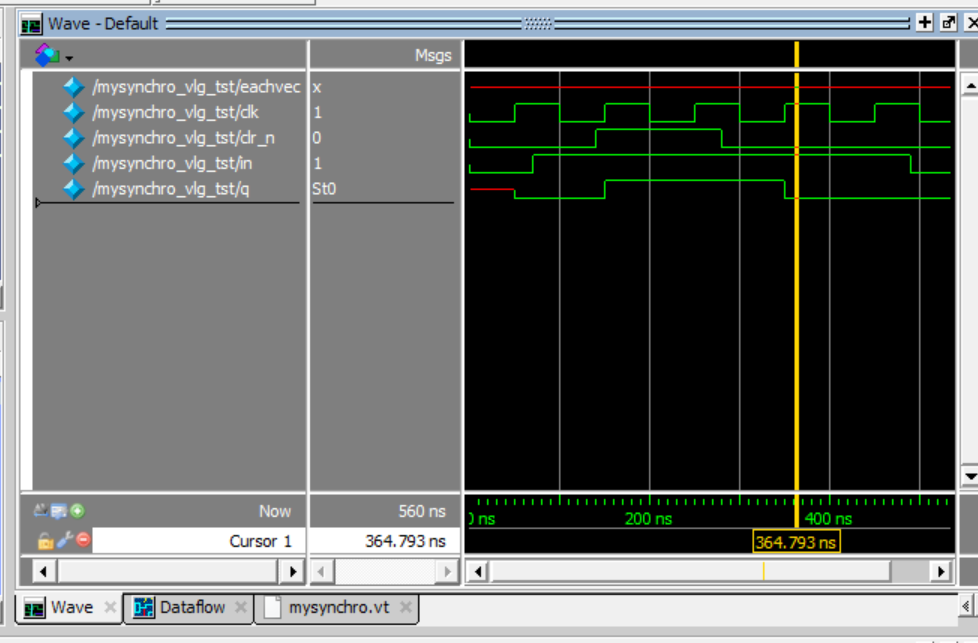
always

begin

#5 clk=~clk;

end

* 仿真结果

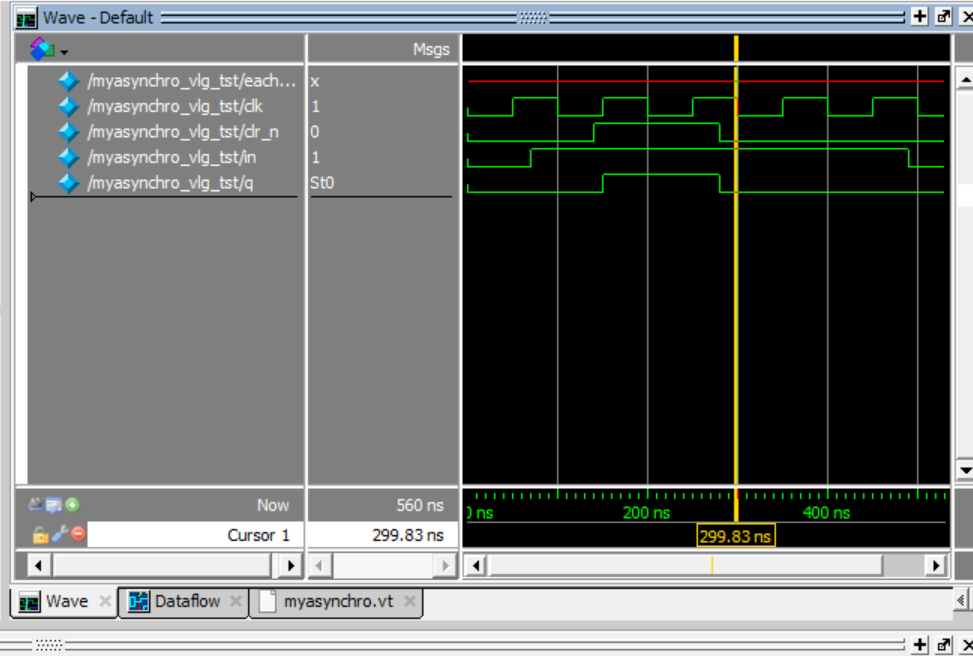


可以看到输出在clr\_n有效后的上升沿才清零

(3)异步清零触发器

Clr\_n有效时立即清零，模块命名为myasynchro

* 测试代码与上述相同
* 仿真结果：



可以看到输出在clr\_n有效后立即清零，没有等待clk的上升沿

（3）顶层模块设计

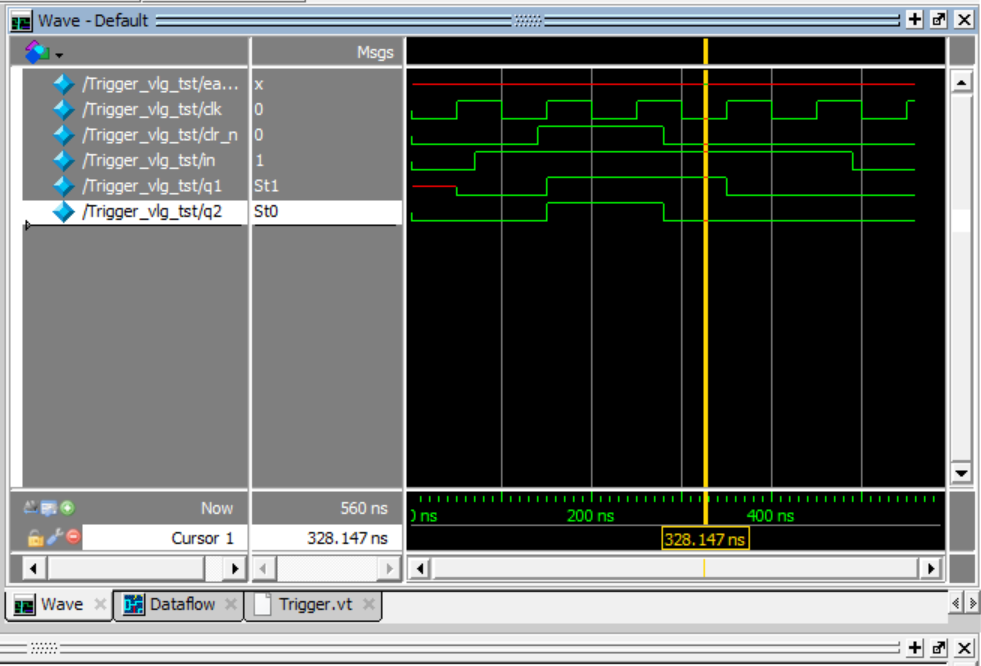
设计Trigger模块

将之前的模块实例化，in时输入，q1和q2是两个触发器的输出

mysynchro lock1(.in(in),.clr\_n(clr\_n),.clk(clk),.q(q1));

myasynchro lock2(.in(in),.clr\_n(clr\_n),.clk(clk),.q(q2));

仿真结果：



六．实验反思和收获

* 实例化时必须给出具体的实例名
* always的触发条件不能同时有边沿触发和电平触发
* 在时序电路中使用<=赋值
* 在quartus中对多模块编写时，要对其中一个模块进行编译和仿真测试，先将其设置为top-lever-entity