

课后作业

本次作业对比时序逻辑always块中，有依赖的两行赋值语句，用阻塞和非阻塞赋值的区别。请分别将以下两份代码在Vivado中进行RTL代码在RTL分析、综合、仿真，下板不做要求，并分别指出RTL分析图、综合后电路图、仿真波形上有哪些异同。

以下两个计数器电路，区别在于一个用非阻塞赋值一个用阻塞赋值。对比仿真和得到的电路差异，并回答：cnt的初始化的值是0，在仿真波形里面，一个时钟后cnt是多少，是1还是6还是5？2个时钟后的cnt是多少？对两个电路结合仿真波形分别作答。

```
module ex1_nonblock(
    input  clk,
    input  rst,
    output [3:0] data
);
    reg [3:0] cnt;
    always @(posedge clk or posedge rst)
    begin
        if(rst) cnt <=0;
        else begin
            cnt <= 5;
            cnt <= cnt + 1'b1;
        end
    end

    assign data = cnt;

endmodule
```

```
module ex1_block(
    input  clk,
    input  rst,
    output [3:0] data
);
    reg [3:0] cnt;
    always @(posedge clk or posedge rst)
    begin
        if(rst) cnt = 0;
        else begin
            cnt = 5;
            cnt = cnt + 1'b1;
        end
    end

    assign data = cnt;

endmodule
```

对应用到的testbench，两个模块的仿真逻辑相同，记得修改例化的模块。

```
`timescale 1ns/1ps
module ex1_tb;
    reg clk;
    reg rst;
    wire [3:0] data;

    ex1_nonblock  u_ex1_nonblock ( // 注意例化的模块与RTL代码一致
        .clk(clk),
        .rst(rst),
        .data(data)
    );

    initial begin
        clk = 1'b0;
        rst = 1'b1;
        #10;
        rst = 1'b0;
    end

    always #5  clk = ! clk;

endmodule
```

🔗 注意

testbench中例化的模块名称注意要与被测的RTL代码一致

确保右上角的综合状态是最新的，再查看综合后的电路图

如果always块的信号之间没有关联，用阻塞和非阻塞赋值生成的电路一样。如果信号之间有关联，两种赋值方式生成的电路区别很大。阻塞和非阻塞赋值很容易用错，要深入理解阻塞和非阻塞赋值的区别需要理解Verilog基于事件的仿真模型。Verilog语法标准中的“执行”指的是Verilog代码仿真时的仿真执行，对于综合后的电路，只有电路的运行，不存在Verilog代码执行的概念。

- 阻塞赋值，仿真时，语句按顺序执行。当前语句完成赋值后，变量立即更新其值，后续语句将使用该更新后的值。
- 非阻塞，仿真时，语句并行执行，块内所有右侧表达式先计算出新值，然后在当前时间步结束时同时更新到左侧变量。因此，语句的执行顺序不会影响最终结果，前一句的赋值不会立即影响后一句中变量的取值。

为了写出符合预期的可综合的代码，请大家遵守以下两条规范

- always块语句建模组合电路，敏感信号列表无时钟，用阻塞赋值=；
- always块语句建模时序电路，敏感信号列表只能有时钟和复位，用非阻塞赋值<=。

尤其注意计数器的实现一定要用对赋值方式，往年很多同学用错了赋值方式，导致计数器不正常工作。