

作业5-分析报告

题目

根据以下核心代码，写出完整模块并生成的电路，进行分析，体会：同步和异步，上升沿和下降沿触发，体会为什么会生成触发器。写成分析报告。

```
1) always @(posedge clk) begin+
    if(!rst_n) begin+
        test_r <= 1;+
    end+
    else begin+
        test_r <= in;+
    end+
end+
```

```
2) always @(negedge clk) begin+
    if(!rst_n) begin+
        test_r <= 1;+
    end+
    else begin+
        test_r <= in;+
    end+
end+
```

```
3) +
always @(posedge clk or negedge rst_n)    begin
    if(!rst_n)    begin
        sd_init_flag_r <= 1'b0;        //复位初始化
    end
    else    begin
        if(in)
            sd_init_flag_r <= 1;        //初始化完成
        end
    end
end
assign    test = sd_init_flag_r;
```

```
4) +
always @(posedge clk or negedge rst_n)    begin
    if(!rst_n)    begin
        sd_init_flag_r <= 1'b0;        //复位初始化
    end
    else    begin
        if(in)
            sd_init_flag_r <= 1;        //初始化完成
        else
            sd_init_flag_r <= 0;        //初始化完成
        end
    end
end
```

基本概念理解

同步，指触发器在受到激励信号激励时，输出还要与时钟信号保持一致，即在时钟信号上升沿或下降沿的作用下置零或置位。

异步，指触发器输出不必与时钟信号一致，只要有激励信号的上升沿或者下降沿的激励，那么它就会置零或者置位。

触发器，在时钟的边沿时刻改变状态的存储元件。上升沿触发，即触发器在时钟的上升沿（0-->1）改变状态。下降沿触发，即触发器在时钟的下降沿（1-->0）改变状态。

四种可以综合出的D触发器：FDCE 同步使能，异步复位；FDRE 同步使能，同步复位；FDPE 同步使能，异步置位；FDSE 同步使能，同步置位。

一、同步上升沿触发的主从D触发器

- 完整模块代码

```
`timescale 1ns / 1ps
module work5_1(
    input rst_n1,in1,clk1,
    output reg test_r1
);
    always@(posedge clk1)           // 上升沿触发的主从D触发器
    begin
        if(!rst_n1)                // 同步置位
        begin
            test_r1 <= 1;           // 非阻塞赋值
        end
        else
        begin
            test_r1 <= in1;
        end
    end
endmodule
```

- RTL电路原理图

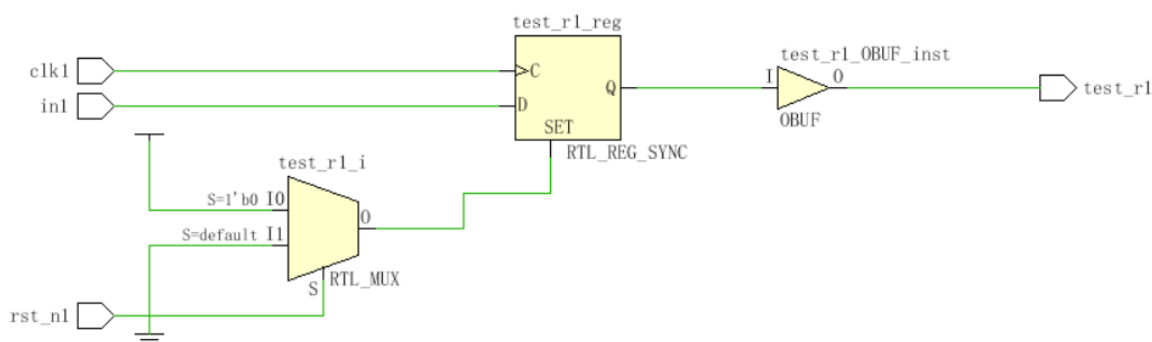


图5-1 RTL原理图

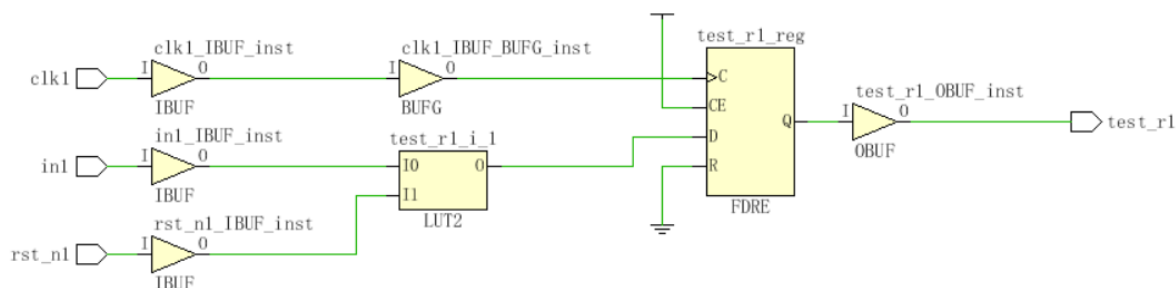


图5-1 Synthesis原理图

- 分析电路

RTL_MUX: 双路选择器，对应代码中的 `if...else...` 模块。

RTL_REG_SYNC: 同步上升沿触发的主从D触发器。always模块敏感列表中有时钟信号 `posedge clk1`，只有一个时钟信号控制，故 `test_r1` 的赋值命令和 `clk1` 的上升沿是同步进行的。寄存器（`reg test_r1`）会在时钟处于上升沿时开始执行，即该存储元件与时序有关，故会生成一个触发器。具体执行分析如下，当时钟 `clk1` 处于上升沿时，`rst_n1` 为0，则 `test_r1` 置位为1；`rst_n1` 为1，则 `test_r1` 置位为in1。随后输出寄存器结果。

LUT: 查找表 (look-up-table)，每写一个逻辑电路，LUT都会事先存储好该电路所有可能的结果，所谓的输入一个（或一组）信号进行逻辑运算，即查找对应的地址，将结果输出即可。

IBUF: 输入缓冲流。

OBUF: 输出缓冲流。

BUFG: 全局缓冲。

FDRE: 同步使能，同步复位。

二、同步下降沿触发的主从D触发器

- 完整代码

```
`timescale 1ns / 1ps
module work5_2(
    input clk2,rst_n2,in2,
    output reg test_r2
);
always @(negedge clk2)      // 下降沿触发的主从D触发器
begin
    if(!rst_n2)              // 多路选择器，同步置位
    begin
        test_r2 <= 1;        // 非阻塞赋值
    end
    else
    begin
        test_r2 <= in2;
    end
end
end
endmodule
```

- 电路原理图

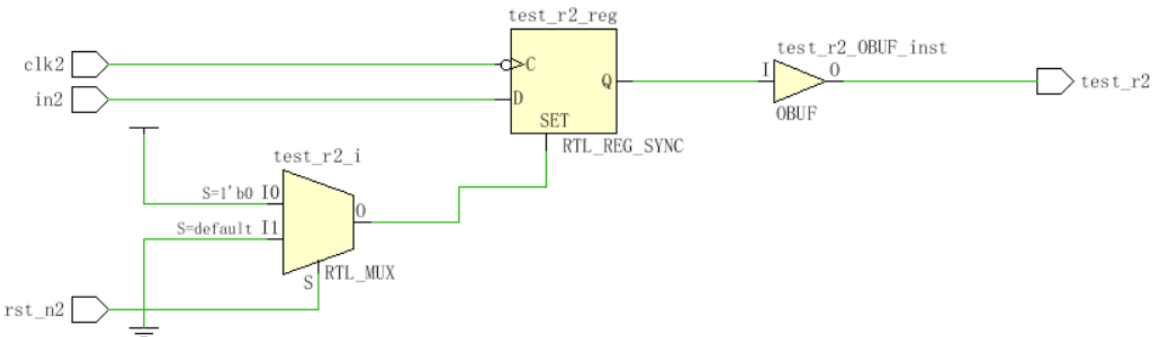


图5-2 RTL原理图

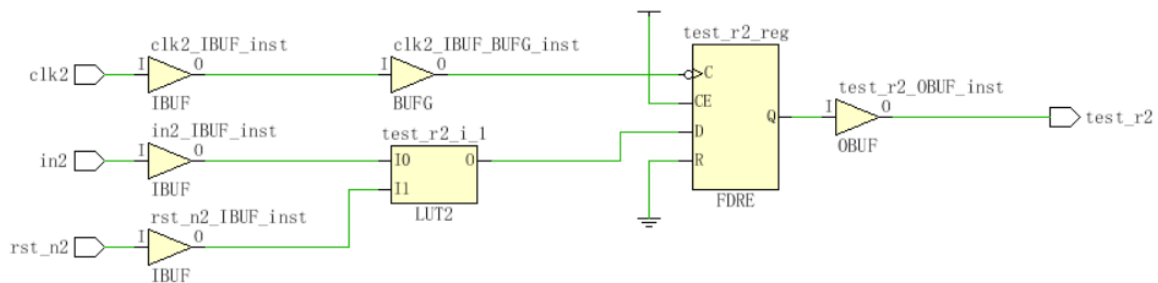


图5-2 Synthesis原理图

- 分析电路

RTL_REG_SYNC: 同步下降沿触发的主从D触发器。always模块敏感列表中有时钟信号 `posedge clk2`，只有一个时钟信号控制，故 `test_r2` 的赋值命令和 `clk2` 的下降沿是同步进行的。寄存器（`reg test_r1`）会在时钟处于下降沿时开始执行，即该存储元件与时序有关，故会生成一个触发器。具体执行分析如下，当时钟 `clk2` 处于下降沿时，`rst_n2` 为0，则 `test_r2` 置位为1；`rst_n2` 为1，则 `test_r2` 置位为in1。随后输出寄存器结果。

FDRE: 同步使能，同步复位。

三、带有异步复位端的上升沿触发的主从D触发器

- 完整代码

```
`timescale 1ns / 1ps
module work5_3(
    input clk3,rst_n3,in3,
    output test3
);
    reg sd_init_flag_r3;
    always @(posedge clk3 or negedge rst_n3) // 敏感事件列表中，同步使能，异步复位
    begin
        if(!rst_n3)
        begin
            sd_init_flag_r3 <= 1'b0; // 当rst_n3=0时，触发器异步复位（复
0)
        end
        else
        begin
            if(in3) // 同步置位（置1）
            sd_init_flag_r3 <= 1;
        end
    end
    assign test3 = sd_init_flag_r3;
endmodule
```

- 电路原理图

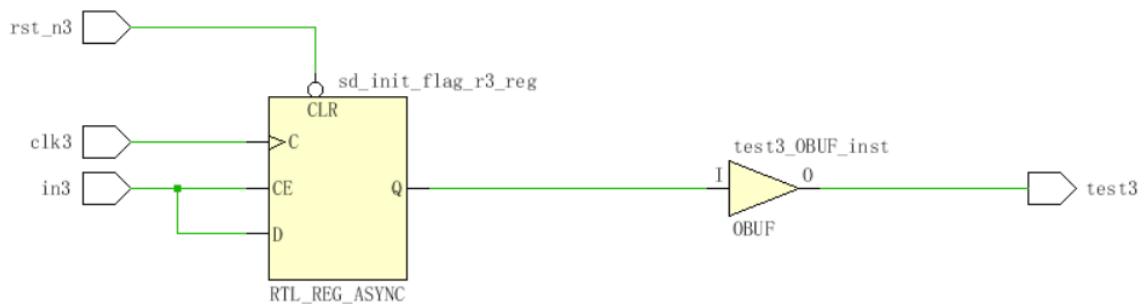


图5-3 RTL原理图

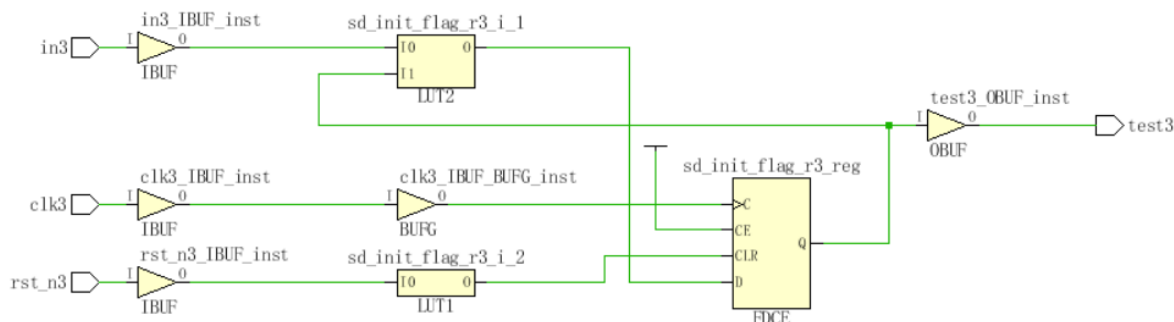


图5-3 Synthesis原理图

- 分析电路

RTL_REG_ASYNC: 带有异步复位端的上升沿触发的主从D触发器。always模块中，敏感列表有两个，`posedge clk3 or negedge rst_n3`，这里的 `rst_n3` 是一个与时钟信号无关的独立的激励信号。寄存器（`sd_init_flag_r3`）在时钟的上升沿（`posedge clk3`）或激励的下降沿（`negedge rst_n3`）开始执行，该存储元件涉及时序，故会生成一个触发器。具体执行分析如下，当激励信号 `rst_n3` 处于下降沿时，寄存器 `sd_init_flag_r3` 复位为0；激励信号处于其他状态时，寄存器会在时钟 `clk3` 的上升沿且 `in3=1` 情况下，同步置位为1。随后输出寄存器结果。

FDCE: D触发器的一种，同步使能，异步置位，具体执行分析同上述内容。

四、带有异步复位端的上升沿触发的主从D触发器

- 完整代码

```
`timescale 1ns / 1ps
module work5_4(
    input clk4,rst_n4,in4,
    output test4
);
    reg sd_init_flag_r4;
    always @(posedge clk4 or negedge rst_n4) // 敏感事件列表中，同步使能，异步
复位
        begin
            if(!rst_n4) // 当rst_n4=0时，触发器异步复位
            (复0)
                begin
                    sd_init_flag_r4 <= 1'b0;
                end
            else
                begin
                    if(in4) // 当in4=1时，同步置位（置1）
                        sd_init_flag_r4 <= 1;
                    else // 当in4=0时，同步置位（置0）
                        sd_init_flag_r4 <= 0;
                    end
                end
            end
end
```

```

        end
    end
    assign test4 = sd_init_flag_r4;
endmodule

```

- 电路原理图

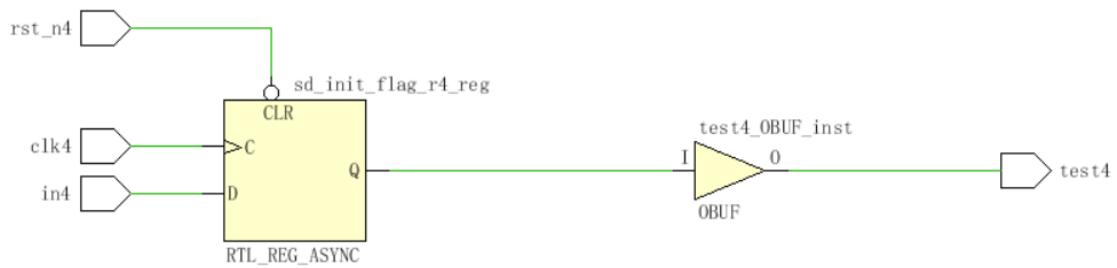


图5-4 RTL原理图

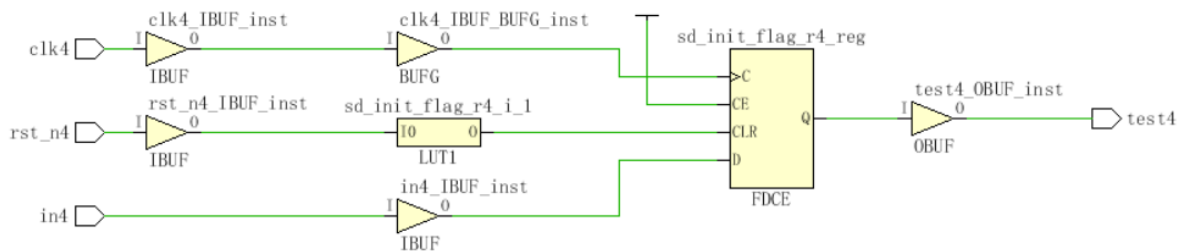


图5-4 Synthesis原理图

- 分析电路

RTL_REG_ASYNC: 带有异步复位端的上升沿触发的主从D触发器。always模块中，敏感列表有两个，`posedge clk4 or negedge rst_n4`，这里的 `rst_n4` 是一个与时钟信号无关的独立的激励信号。寄存器 (`sd_init_flag_r4`) 在时钟的上升沿 (`posedge clk4`) 或激励的下降沿 (`negedge rst_n4`) 开始执行，该存储元件涉及时序，故会生成一个触发器。具体执行分析如下，当激励信号 `rst_n4` 处于下降沿时，寄存器 `sd_init_flag_r4` 复位为0；激励信号处于其他状态时，寄存器会在时钟 `clk4` 的上升沿同步置位，`in4=1`时置1，`in4=0`时置0。随后输出寄存器结果。

FDCE: 同步使能，异步置位。