

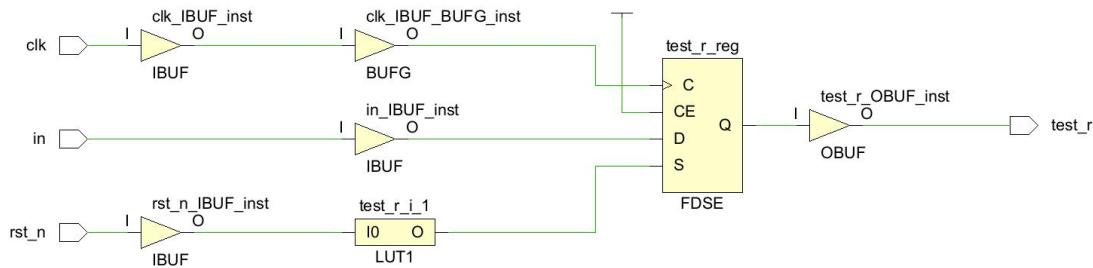
## 作业六

1)

类型：上升沿触发同步复位电路

代码：

```
module dff_posedge (
    input clk,
    input rst_n,
    input in,
    output reg test_r
);
    always @(posedge clk)begin
        if(!rst_n)begin
            test_r <=1;
        end;
        else begin
            test_r <=in;
        end
    end
Endmodule
```

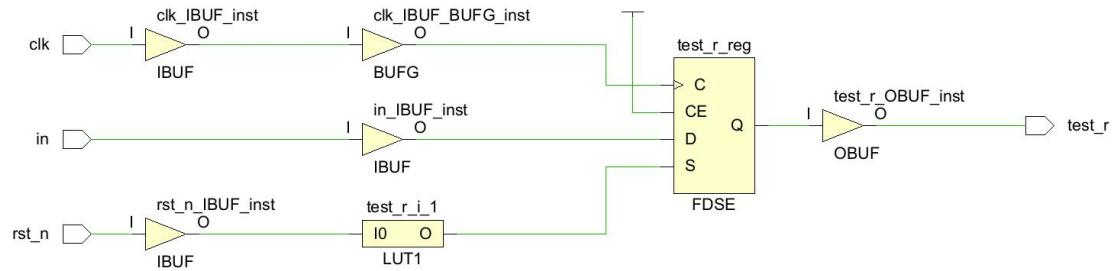


2)

类型：下降沿触发 D 触发器

代码：

```
module dff_negedge (
    input clk,
    input rst_n,
    input in,
    output reg test_r
);
    always @(negedge clk) begin
        if(!rst_n)
            test_r <= 1'b1;
        else
            test_r <= in;
    end
endmodule
```

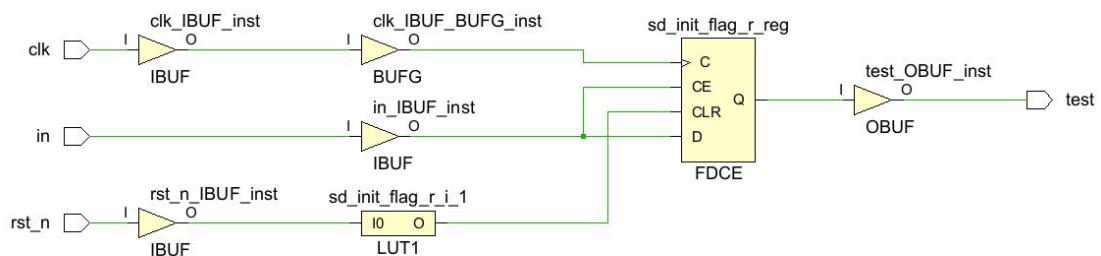


3)

类型：异步复位（低电平有效）上升沿触发电路

代码：

```
module y_posedge_mode(
    input clk,
    input rst_n,
    input in,
    output test );
    reg sd_init_flag_r;
    always @(posedge clk or negedge rst_n) begin
        if(!rst_n) begin
            sd_init_flag_r <= 1'b0;
        end
        else begin
            if(in) begin
                sd_init_flag_r <= 1;
            end
        end
    end
    assign test = sd_init_flag_r;
Endmodule
```



4)

类型：异步复位（低电平有效）上升沿触发带 else 分支电路

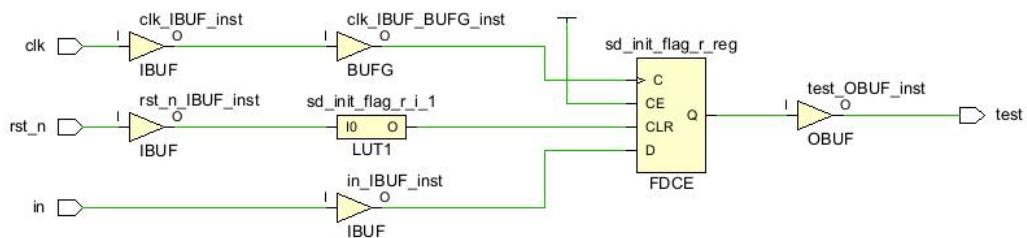
代码：

```
module y_posedge_mode(
    input clk,
    input rst_n,
```

```

input in,
output test );
reg sd_init_flag_r;
always @(posedge clk or negedge rst_n) begin
if(!rst_n) begin
sd_init_flag_r <= 1'b0;
end
else begin
if(in) begin
sd_init_flag_r <= 1;
end
else begin
sd_init_flag_r <= 0;
end
end
end
assign test = sd_init_flag_r;
Endmodule

```



生成触发器的原因：

always 块的触发条件是 时钟沿触发（posedge/negedge）；  
且使用了 非阻塞赋值 (<=)；  
满足 D 触发器的行为描述形式（在时钟沿采样输入、保持输出）；  
因此综合工具会自动将其识别为一个 D 触发器（Flip-Flop）。