1.（1）代码：

对原模块补齐后设计文件如下图所示，为方便区分和编辑将模块名命名为m1。输入输出分别为时钟控制变量 clk，复位输入 rst_n，输入 in 和输出 test_r。

```
module m1(clk,in,rst_n,test_r);
input clk,in,rst_n;
output reg test_r;
always @(posedge clk) begin
    if(!rst_n) begin
        test_r <= 1;
    end
    else begin
        test_r <= in;
    end
end
endmodule
```

设计文件代码

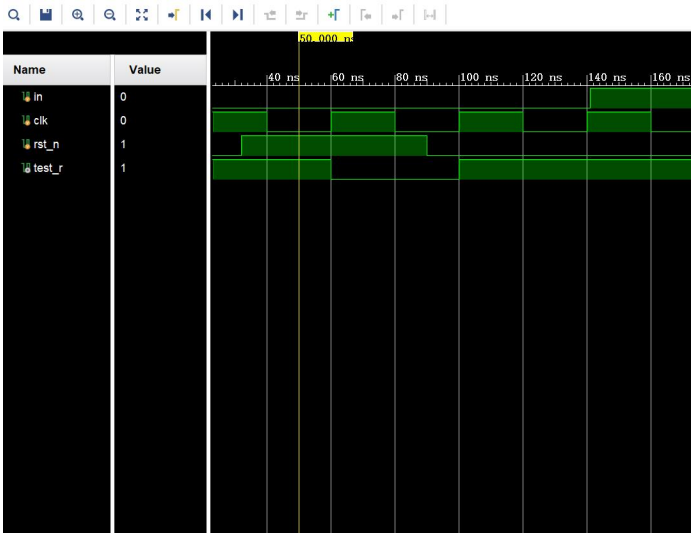
```
module sim_m(

    );
    reg in=0,clk=0,rst_n=0;
    wire test_r;
    m1 u1(clk,in,rst_n,test_r);
    initial begin
        forever #20 clk=~clk;
    end
    initial begin
        in=0;
        #32 rst_n=1;
        #58 rst_n=0;
        #51 in=1;
        #43 rst_n=1;
    end
endmodule
```

仿真文件代码

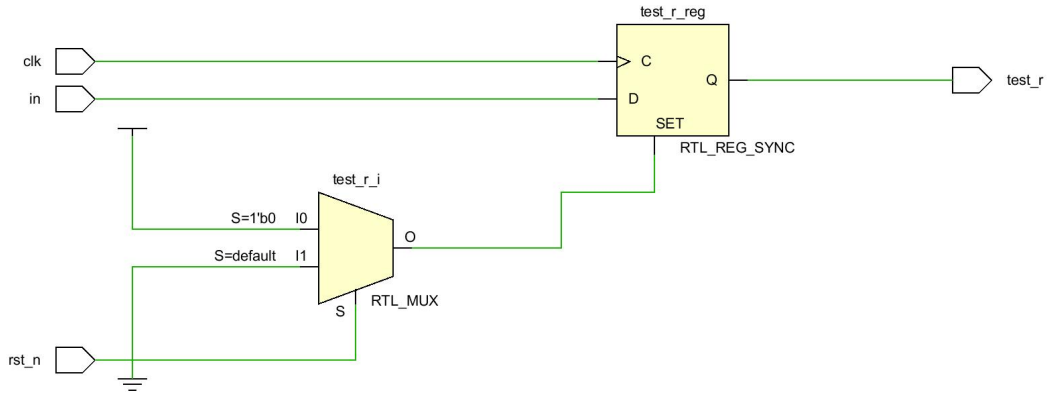
（2）仿真结果

新建项目，新建设计文件和仿真文件，分别命名为 m1 和 sim_m。将上述代码输入，运行行为仿真，得到如下结果，以便后续分析。



（3）生成电路

利用 vivado 软件自带的 RTL 分析功能分析设计代码所形成的模拟电路图，如下图所示。



（4）实验分析

该代码的主要功能是在每一个时钟的**上升沿**时：当复位输入置一时，输出 test_r 由输入 in 决定；当复位输入置零时，输出 test_r 复位，置一。是一个**同步时序电路**。

仿真文件将时钟的触发周期设为 20，同时将各变量的变化周期设置为 20 的非整数倍，以便观察当时钟的上升沿之前改变变量值对输出的影响。仿真结果显示，当改变变量值时，输出在时钟上升沿才对应地改变值。

由生成的电路可以看出，生成了一个带有 D 触发器的电路。这是由时序逻辑

电路需要在时钟边沿触发的条件决定的。触发器的结构使他能够保存数据，保存电路状态，而时序电路需要在时钟边沿触发，所以电路生成了触发器的结构，而题中的逻辑要求正好符合 D 触发器的输入输出逻辑。

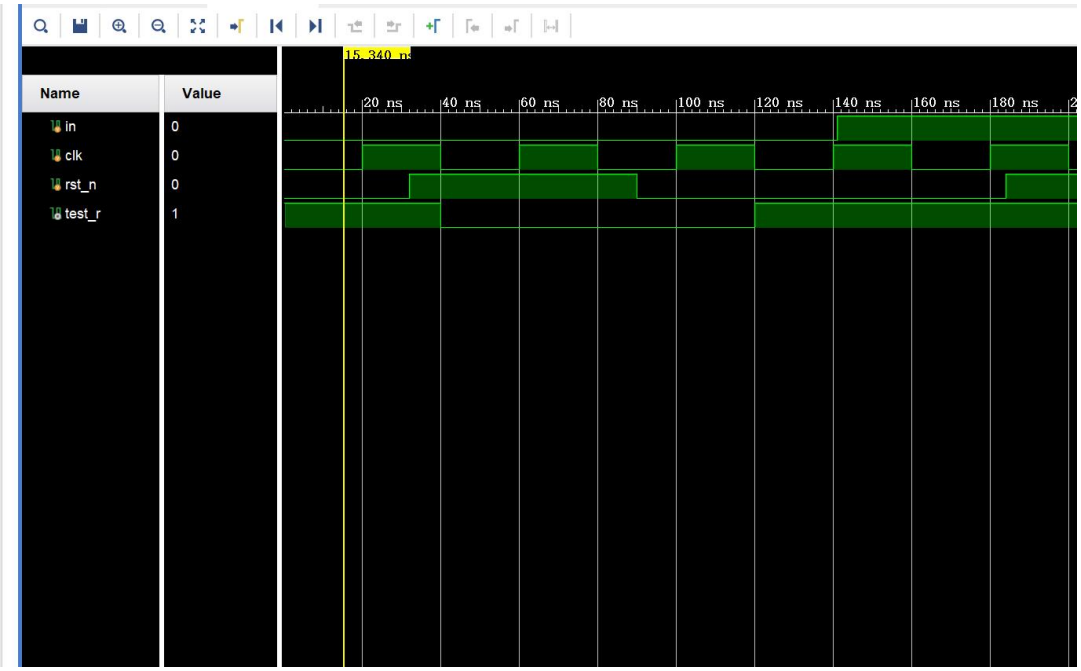
2.（1）代码

本题的代码模块与之前唯一的区别是 always 中的敏感信号表达式，时钟变化从上升沿触发变成了下降沿触发，其他不变。因此在设计文件的编写上与上述只有这一个变化，如下图所示。同时仿真文件除了实例化调用时将模块名从 m1 改到 m2 外不变，观察仿真输出结果。

```
module m2(clk,in,rst_n,test_r);
input clk,in,rst_n;
output reg test_r;
always @(negedge clk) begin
    if(!rst_n) begin
        test_r <= 1;
    end
    else begin
        test_r <= in;
    end
end
endmodule
```

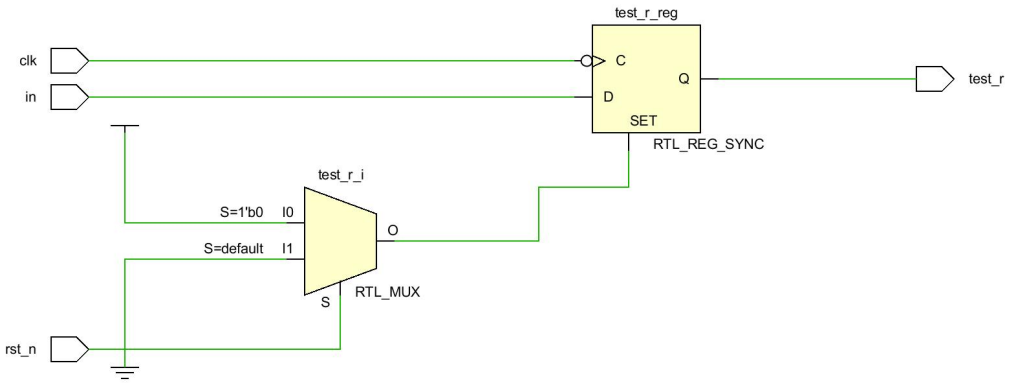
（2）仿真结果

新建项目，新建设计文件和仿真文件，分别命名为 m2 和 sim_m。将上述代码输入，运行行为仿真，得到如下结果，以便后续分析。



（3）生成电路

利用 vivado 软件自带的 RTL 分析功能分析设计代码所形成的模拟电路图，如下图所示。



（4）实验分析

该代码与 1 中的代码类似，主要功能是在每一个时钟的**下降沿**时：当复位输入置一时，输出 test_r 由输入 in 决定；当复位输入置零时，输出 test_r 复位，置一。是一个**同步时序电路**。

仿真文件将时钟的触发周期设为 20，同时将各变量的变化周期设置为 20 的非整数倍，以便观察当时钟的上升沿之前改变变量值对输出的影响。仿真结果显

示，当改变变量值时，输出在时钟下降沿才对应地改变值。

由生成的电路可以看出，该代码也生成了一个带有 D 触发器的电路。

该代码在与 1 中的代码区别最大的是，该代码的触发方式是时钟下降沿触发，只有当时钟处于下降沿时才会执行 always 模块里的要求。

3.（1）代码

对原模块补齐后设计文件如下图所示，为方便区分和编辑将模块名命名为 m3。输入输出分别为时钟控制变量 clk，复位输入 rst_n，输入 in 和输出 test。

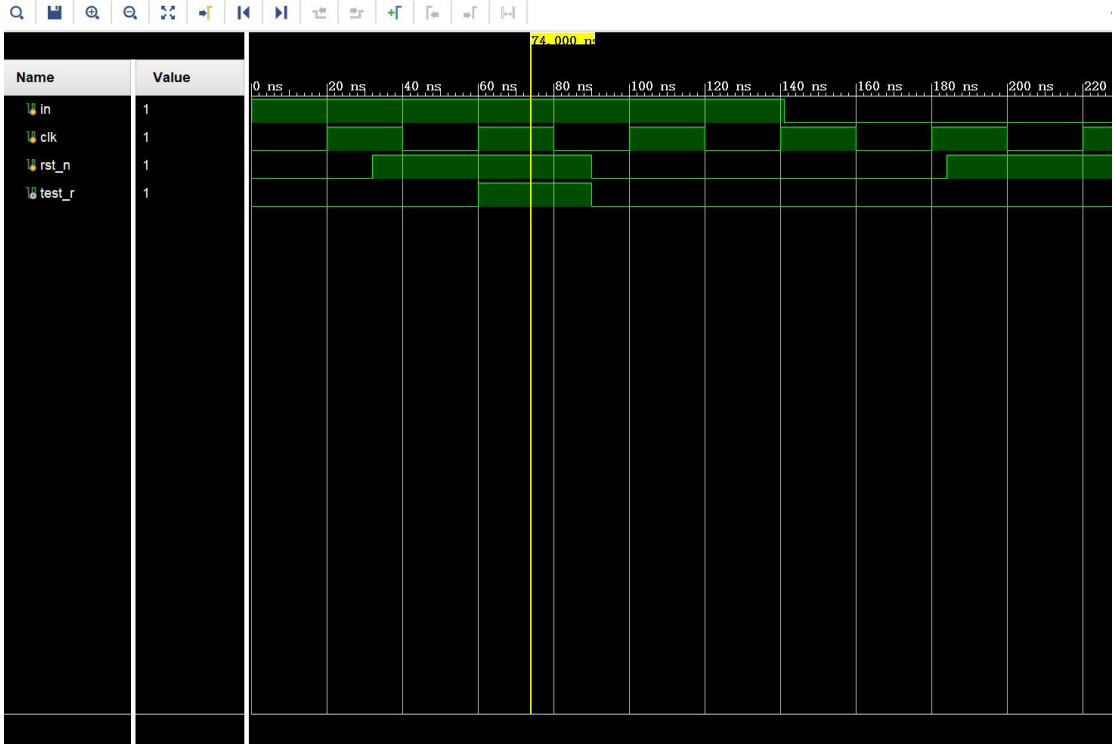
```
module m3(clk,in,rst_n,test);
input clk,in,rst_n;
reg sd_init_flag_r;
output test;
always @(posedge clk or negedge rst_n) begin
    if(!rst_n) begin
        sd_init_flag_r <= 1'b0;
    end
    else begin
        if(in)
            sd_init_flag_r <= in;
    end
end
assign test=sd_init_flag_r;
endmodule
```

```
module sim_m(

    );
    reg in=1,clk=0,rst_n=0;
    wire test_r;
    m3 u1(clk,in,rst_n,test_r);
    initial begin
        forever #20 clk=~clk;
    end
    initial begin
        in=1;
        #32 rst_n=1;
        #58 rst_n=0;
        #51 in=0;
        #43 rst_n=1;
    end
endmodule
```

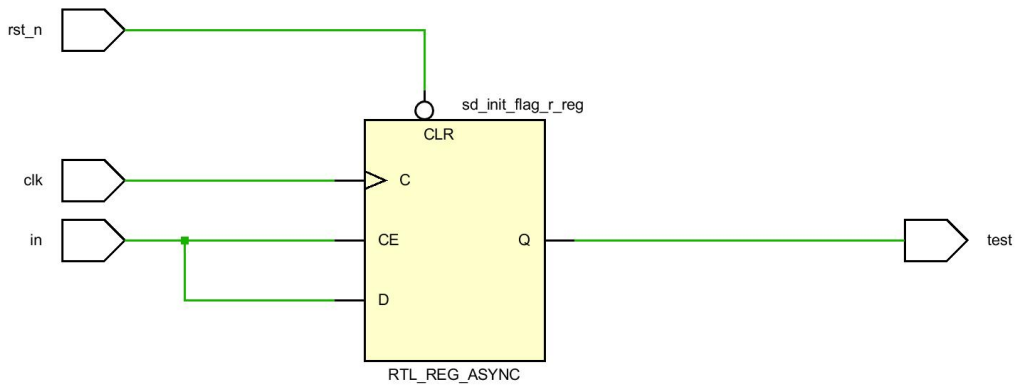
（2）仿真结果

新建项目，新建设计文件和仿真文件，分别命名为 m1 和 sim_m。将上述代码输入，运行行为仿真，得到如下结果，以便后续分析。



（3）生成电路

利用 vivado 软件自带的 RTL 分析功能分析设计代码所形成的模拟电路图，如下图所示。



（4）实验分析

该代码的主要功能是在每一个时钟的**上升沿**和复位输入端 rest_n 的**下降沿**时：当复位输入置一且输入 in 为 1 时，输出 test_r 由输入 in 决定，即输出 1；当复位输入置零等其他情况时，输出 test_r 复位，置零。是一个**异步时序电路**。

仿真文件将时钟的触发周期设为 20，同时将各变量的变化周期设置为 20 的非整数倍，以便观察当时钟的上升沿之前改变变量值对输出的影响。仿真结果显示，当改变变量值时，输出在时钟上升沿及复位输入下降沿才对应地改变值。

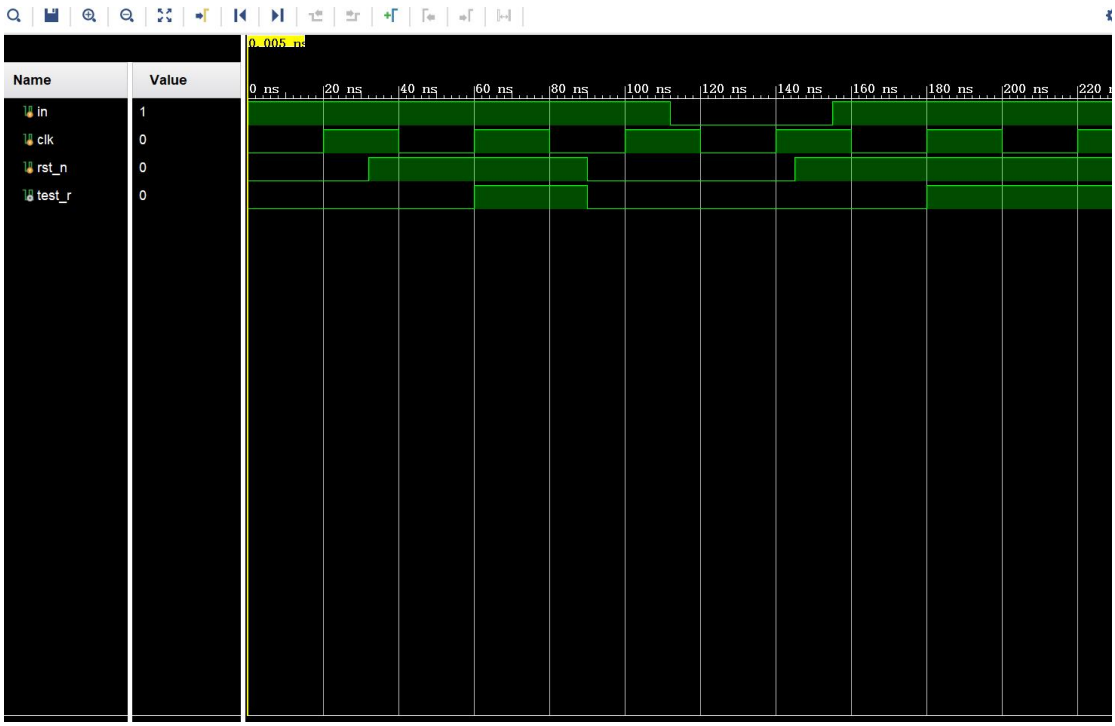
由生成的电路和查阅相关资料，得到生成了一个带使能功能的异步清除 D 触发器。这同样是由异步时序逻辑电路需要在时钟边沿触发的条件决定的。触发器的结构能够保存数据，保存电路状态契合了时序电路需要在时钟边沿触发的特点；而异步时序电路和其他相关的逻辑要求正好符合带使能功能的异步清除 D 触发器的输入输出逻辑。所以生成了这样的电路。

4.（1）代码

本题的代码模块与题 3 的区别是 always 块中的命令行的变化，其他不变，如下图所示。设计文件与题 3 一致。

```
module m4(clk,in,rst_n,test);
input clk,in,rst_n;
reg sd_init_flag_r;
output test;
always @(posedge clk or negedge rst_n) begin
    if(!rst_n) begin
        sd_init_flag_r <= 1'b0;
    end
    else begin
        if(in)
            sd_init_flag_r <= 1;
        else
            sd_init_flag_r <= 0;
    end
end
assign test=sd_init_flag_r;
endmodule
```

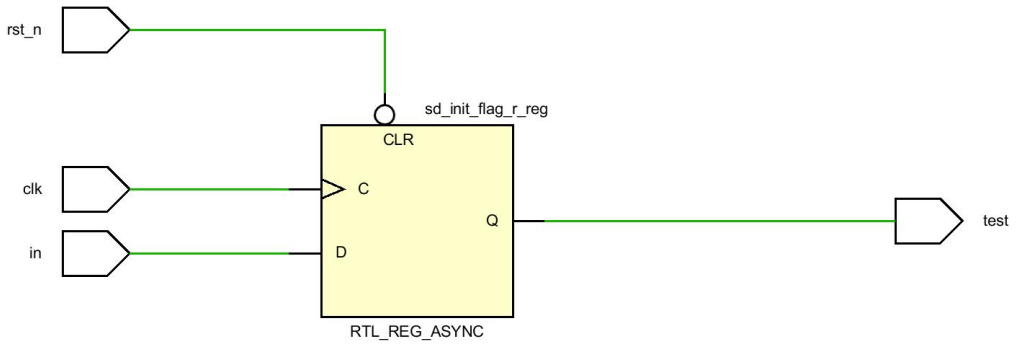
（2）仿真结果



（3）生成电路

利用 vivado 软件自带的 RTL 分析功能分析设计代码所形成的模拟电路图，如下

图所示。



（4）实验分析

该代码的主要功能是在每一个时钟的**上升沿**和复位输入端 rest_n 的**下降沿**时：当复位输入置一时，输出 test_r 由输入 in 决定，即输出 1；当复位输入置零等其他情况时，输出 test_r 复位，置零。是一个**异步时序电路**。

仿真文件将时钟的触发周期设为 20，同时将各变量的变化周期设置为 20 的非整数倍，以便观察当时钟的上升沿之前改变变量值对输出的影响。仿真结果显示，当改变变量值时，输出在时钟上升沿及复位输入下降沿才对应地改变值。

该代码与题 3 代码的区别是输入端 in 起到的功能不同，题 3 中只有复位输入为 1、in 为 1 时输出才是 1，而该代码中复位输入为 1 时，输出和输入端 in 逻辑值等价。这在仿真结果中看起来似乎没有什么区别，但是在实现逻辑上是不同的，具体可以在 vivado 生成的电路中看出。

由生成的电路和查阅相关资料，得到生成了一个带异步清除 D 触发器。这同样是由异步时序逻辑电路需要在时钟边沿触发的条件决定的。触发器的结构能够保存数据，保存电路状态契合了时序电路需要在时钟边沿触发的特点；而异步时序电路和其他相关的逻辑要求正好符合带异步清除 D 触发器的输入输出逻辑。所以生成了这样的电路。