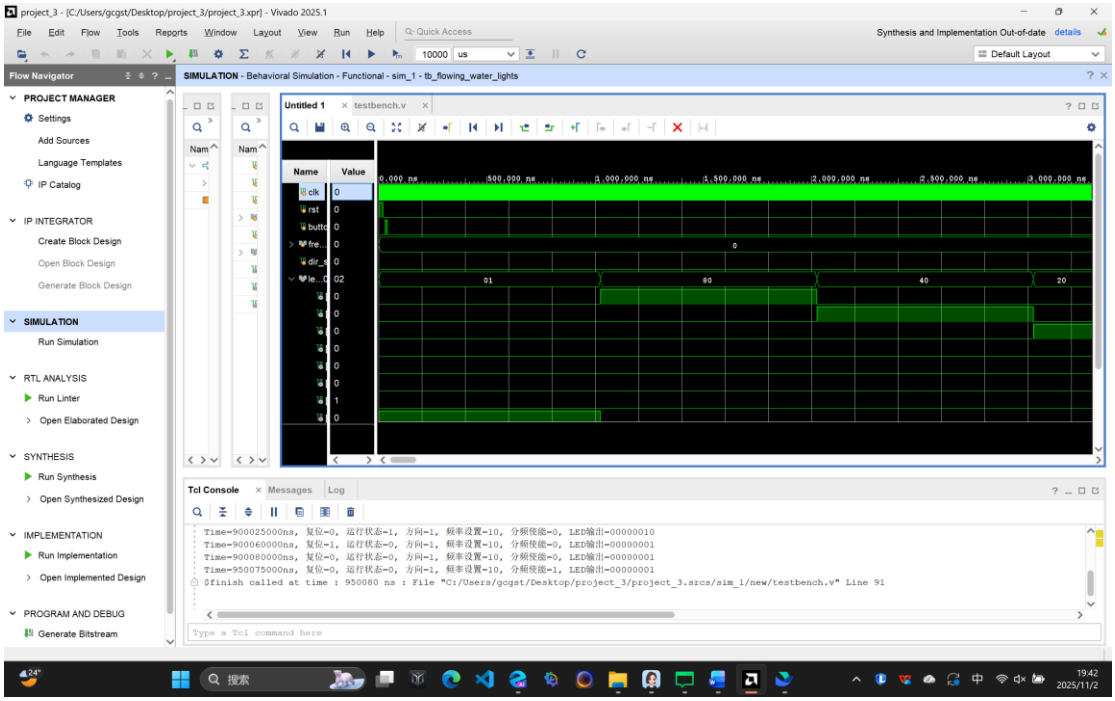
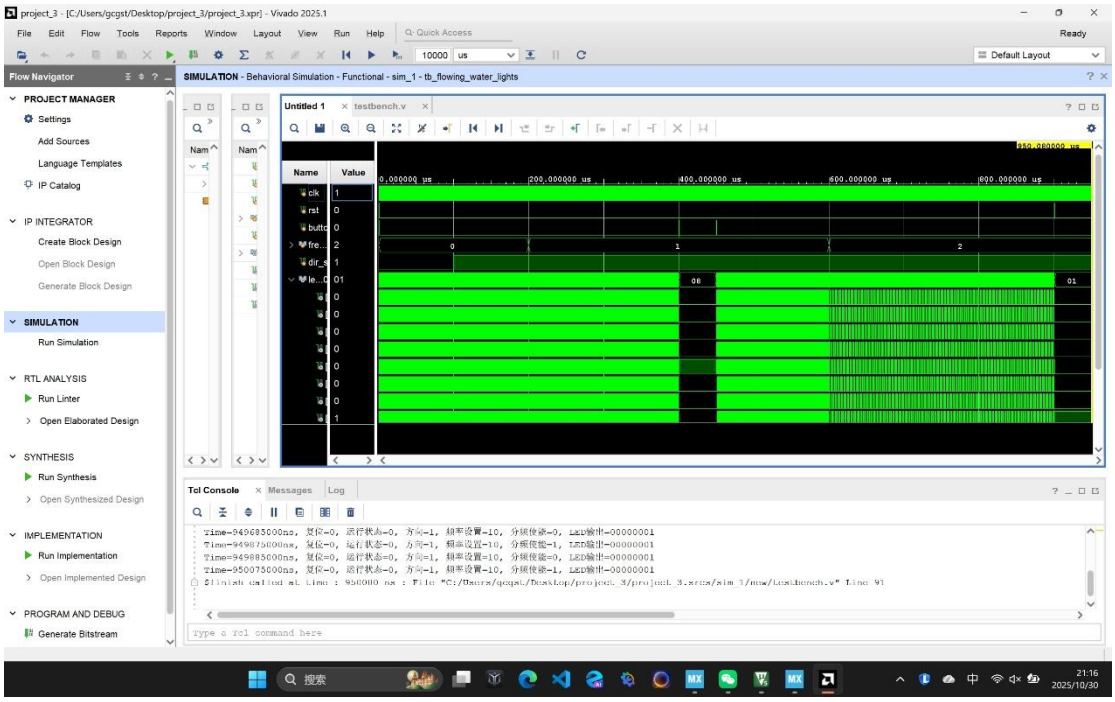
实验 3：计数器

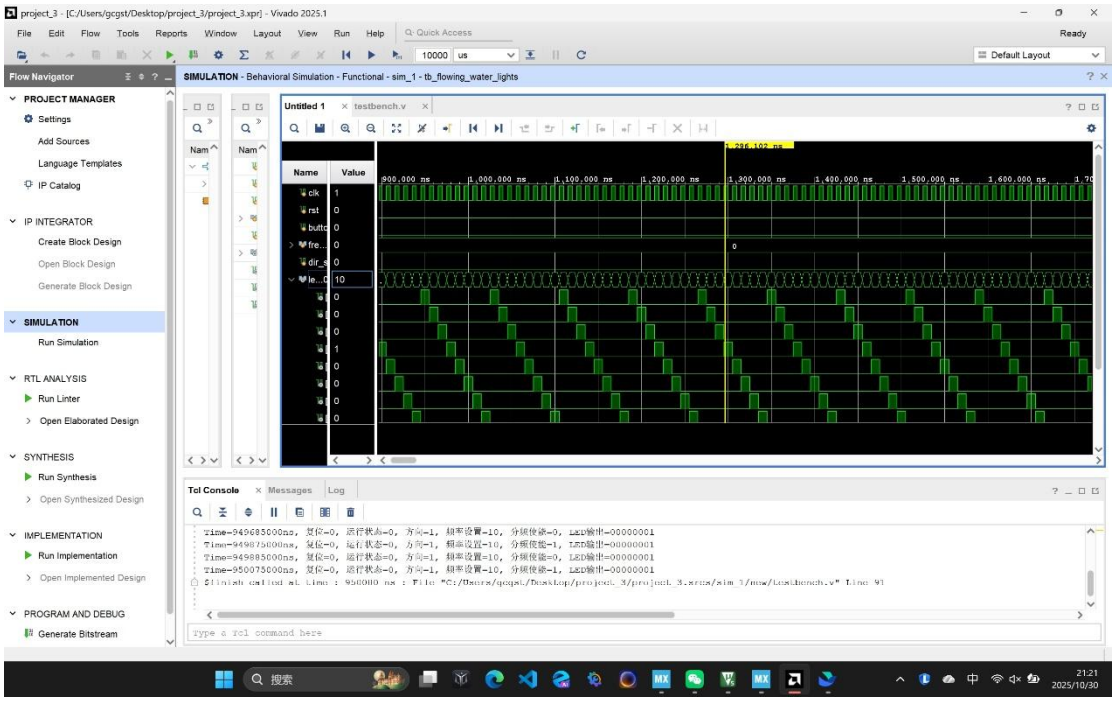
一、基于计数器的流水灯仿真分析

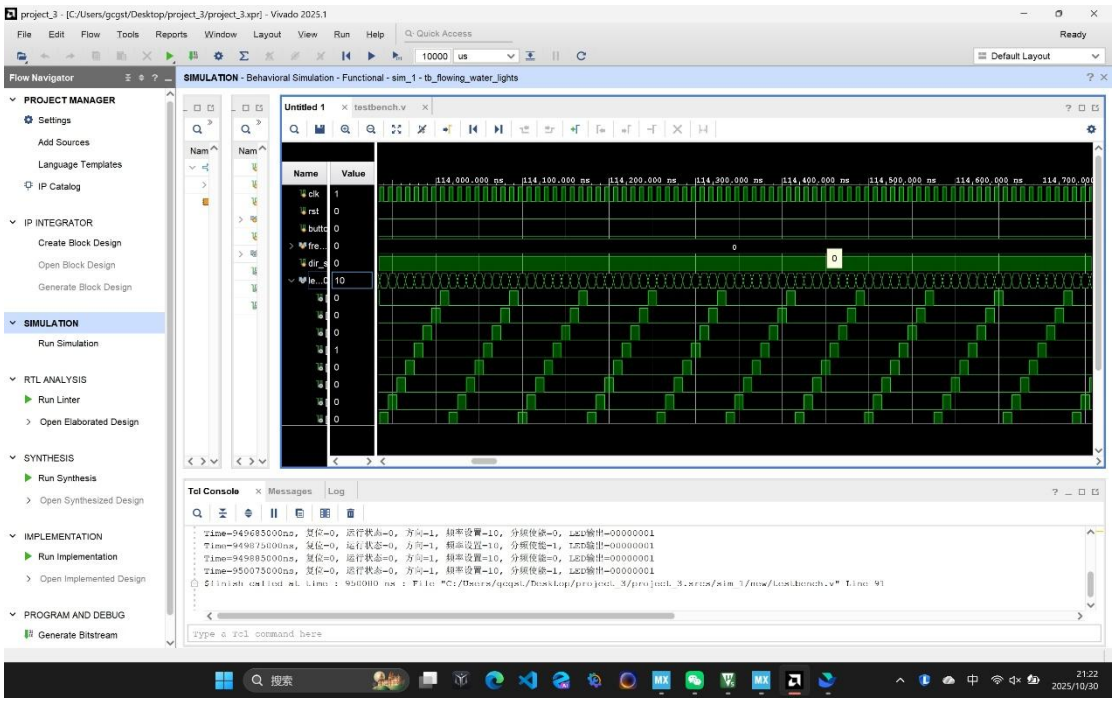
1.1 仿真波形截图

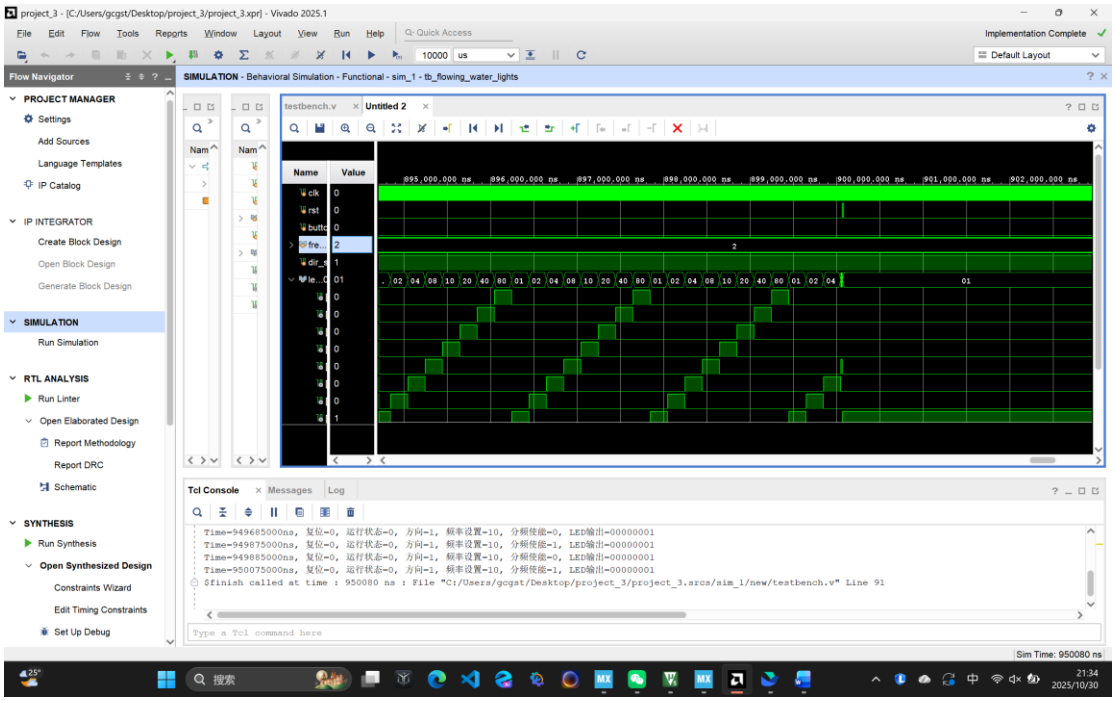
能正确清晰体现实验所要求的功能，根据需要可以截多张仿真波形图
并体现计数器或者输出的 led 信号在某个频率下的周期测量






project_3 - [C:/Users/gcgst/Desktop/project_3/project_3.xpr] - Vivado 2025.1
File Edit Flow Tools Reports Window Layout View Run Help
Quick Access
Ready
Default Layout
Flow Navigator
SIMULATION - Behavioral Simulation - Functional - sim_1 - tb_flowing_water_lights
PROJECT MANAGER
Settings
Add Sources
Language Templates
IP Catalog
IP INTEGRATOR
Create Block Design
Open Block Design
Generate Block Design
SIMULATION
Run Simulation
RTL ANALYSIS
Run Linter
Open Elaborated Design
SYNTHESIS
Run Synthesis
Open Synthesized Design
IMPLEMENTATION
Run Implementation
Open Implemented Design
PROGRAM AND DEBUG
Generate Bitstream
Untitled 1
testbench.v
Name
Value
clk
rst
Tcl Console
Messages
Log
Type a Tcl command here
搜索
21:21
2025/10/30


project_3 - [C:/Users/gcgst/Desktop/project_3/project_3.xpr] - Vivado 2025.1
File Edit Flow Tools Reports Window Layout View Run Help
Quick Access
Ready
Default Layout
Flow Navigator
SIMULATION - Behavioral Simulation - Functional - sim_1 - tb_flowing_water_lights
PROJECT MANAGER
Settings
Add Sources
Language Templates
IP Catalog
IP INTEGRATOR
Create Block Design
Open Block Design
Generate Block Design
SIMULATION
Run Simulation
RTL ANALYSIS
Run Linter
Open Elaborated Design
SYNTHESIS
Run Synthesis
Open Synthesized Design
IMPLEMENTATION
Run Implementation
Open Implemented Design
PROGRAM AND DEBUG
Generate Bitstream
Untitled 1
testbench.v
Name
Value
clk
rst
Tcl Console
Messages
Log
Type a Tcl command here
搜索
21:22
2025/10/30



1.2 波形分析

需体现初始复位、启动、暂停、间隔切换、方向切换，间隔和方向切换只需要体现一次。

图 1:

初始复位:

0-20ns：rst=1，复位。30-40ns：button=1，启动。

图 2:

启动:

400us 时 button=1。按下 button 前 led[3]亮，按下 button 后流水灯停止流动，保持只有 led[3]亮的状态。

暂停:

450us 时 button=1。流水灯继续流动.

间隔切换:

0-200us 时，freq_set=0;

200-400us 时，freq_set=1

400-600us 时，freq_set=2 从而以不同频率输出，流水灯以不同频率流动。

图 3:（图 1 左侧的放大）

dir=0，流水灯右移位

图 4:

dir=1，流水灯左移位

图 5:

900us 时，按下 reset，reset=1，只有 led[0]亮。

二、关键代码

尽管已经提交了代码文件，仍要求给出以下关键代码及说明，其他模块代码无需粘贴，注意排版，最好直接截图

2.1 3 个寄存器级联实现边沿检测的代码

```v
`timescale 1ns / 1ps
module edge_detect(
    input wire clk,
    input wire rst,
    input wire signal,
    output wire pos_edge
);

    reg sig_r0, sig_r1, sig_r2;

always @(posedge clk or posedge rst)
begin
    if(rst) sig_r0 <= 1'b0;
    else    sig_r0 <= signal;
end

always @(posedge clk or posedge rst)
begin
    if(rst) sig_r1 <= 1'b0;
    else    sig_r1 <= sig_r0;
end

always @(posedge clk or posedge rst)
begin
    if(rst) sig_r2 <= 1'b0;
    else    sig_r2 <= sig_r1;
end

assign pos_edge = sig_r1 & ~ sig_r2;

endmodule
```

2.2 按键 S2 启停的实现的代码

贴出关键代码，并简要说明按键 S2 对应的信号如何控制计数器启停

```v
    always @(posedge clk or posedge rst) begin
        if (rst) begin
            running <= 1'b0;
        end else if (pos_edge_button) begin
            running <= ~running;
        end
    end

    always @(posedge clk or posedge rst) begin
        if (rst) begin
            led <= 8'b00000001;
        end else if (clk_en && running) begin
            if (dir_set) begin

                led <= {led[6:0], led[7]};
            end else begin

                led <= {led[0], led[7:1]};
            end
        end
    end
```

按键 S2 通过上升沿触发启停状态切换，最终由 running 寄存器的状态决定计数器（流水灯）是否运行。

步骤 1：边缘检测

调用 edge_detect 对 button 上升沿检测，输出 pos_edge_button，避免按键机械抖动的影响，确保只有按键按下瞬间的上升沿被识别。

步骤 2：启停状态切换

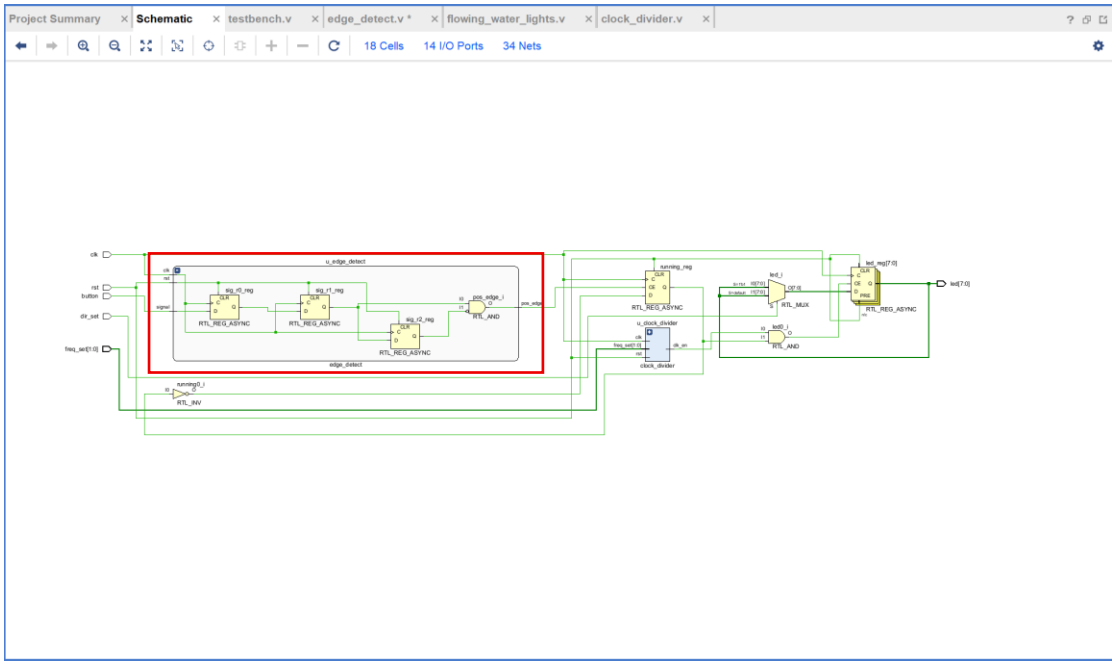
当检测到 pos_edge_button（按键 S2 的上升沿）时，寄存器 running 会翻转状态，1 变为 0 或者 0 变为 1。running=1 表示“允许计数器运行”，running=0 表示“禁止计数器运行”。

步骤 3：控制计数器使能

在流水灯的时序逻辑中，只有当分频时钟使能 clk_en 和 running=1 同时满足时，流水灯才会执行移动操作。

三、流水灯 RTL Analysis 截图

需用红框准确标记边沿检测三级寄存器级联的位置



四、计数器最大值的计算

时钟频率 100MHz，给出流水灯某一个频率对应的计数器变量 cnt 应达到的最大值的计算过程，cnt 从 0 开始，流水灯频率自行选定。

设流水灯频率 $f_{led} = 1000\ Hz = 1\times10^{3} Hz$

系统时钟频率 $f_{clk} = 100MHz = 1\times10^{8} Hz$

系统时钟周期 $T_{clk} = \frac{1}{f_{clk}} = 1\times10^{-8} s$

流水灯切换周期 $T_{led} = \frac{1}{f_{led}} = 1\times10^{-3} s$

$N = \frac{T_{led}}{T_{clk}} = \frac{1\times10^{-3} s}{1\times10^{-8} s} = 1\times10^{5}$

∴计数器最大值 $cnt_{max} = N-1 = 99999$

来自华为笔记