PROPOSAL REPORT

4位计数器设计与仿真



CONTENTS
目录

01 计数器基础概念

02 TD工程创建

03 Verilog代码实现

04 ModelSim仿真流程

05 波形验证与分析



PART

01

计数器基础概念

PART

计数器是什么

计数器定义

计数器是FPGA中基础且重要的时序逻辑电路，其核心功能是对输入的时钟脉冲或特定事件进行计数。它由寄存器存储计数状态，通过组合逻辑实现计数逻辑，如加1、减1或清零等操作。



4位计数器特征

位宽与状态

4位计数器输出位宽为4，可表示0到15共16个状态。每个状态由4个触发器存储，对应4位二进制数。



循环计数

当计数值达到4'b1111即十进制15时，下一时钟沿自动清零并重新开始计数，形成无缝循环，适用于多种周期性任务。



PART

02

TD工程创建

PART

新建TD工程

启动与选择

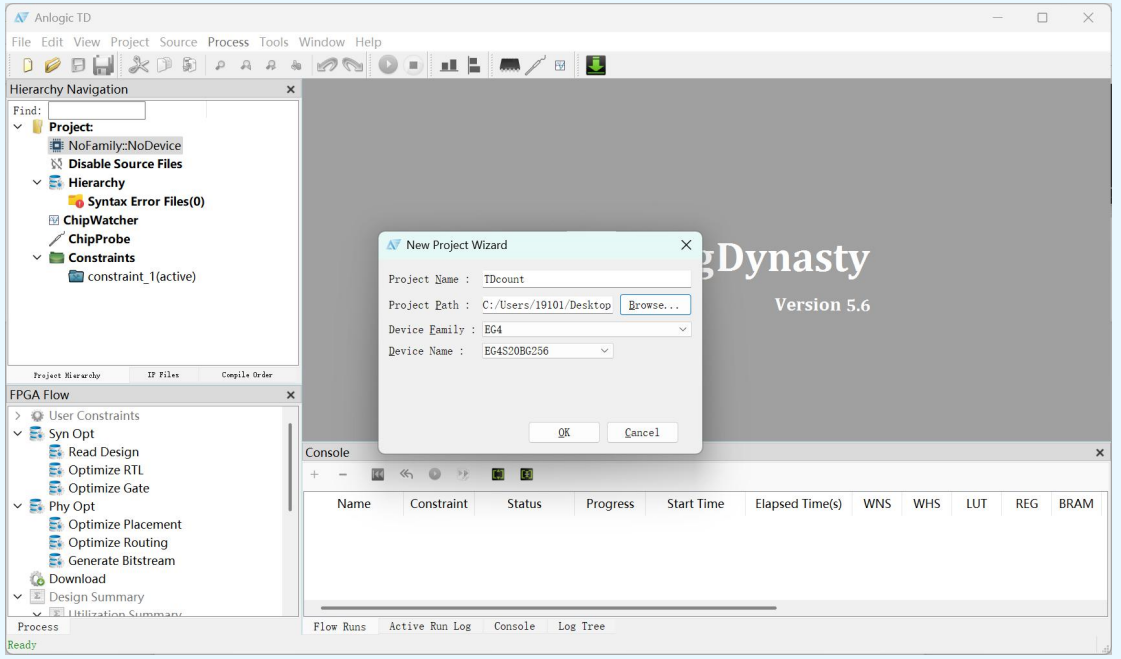
启动安路TD软件，选择对应EAGLE器件型号，建立空白工程。在向导中设置工程名称与保存路径，确保后续生成的比特流可直接下载到目标板。

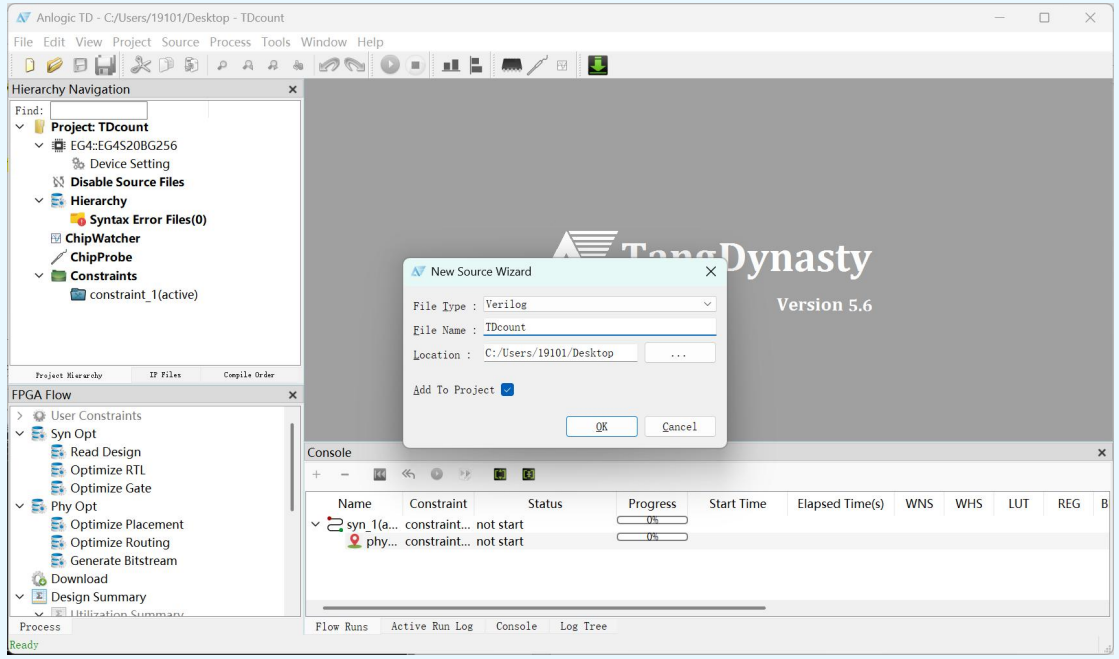
工程窗口

创建完成后，工程窗口将显示空白的源文件列表，为添加计数器设计文件做好准备。

工程准备

此时工程已具备基本框架，可进行后续的设计文件添加与编译操作。





添加设计源文件



新建与添加

在TD工程内新建Verilog文件，命名为Tdcount.v并加入工程。文件内首先声明模块名Tdcount，随后定义输入端口clk、rst_n及4位输出count。通过parameter声明count_max=4'b1111，便于后续修改计数上限。保存后，工程树将自动列出该文件，等待后续编译。



PART

03

Verilog代码实现

PART



端口与参数定义

端口声明

模块接口仅包含三个信号：时钟输入clk、低有效异步复位rst_n、4位计数输出count。端口声明采用ANSI风格，简洁易读。

参数定义

使用parameter定义常量count_max=4'b1111，使计数上限可配置，便于后续修改与扩展。

核心计数逻辑



always块

在always块中检测posedge clk或negedge rst_n，实现异步复位与同步计数逻辑。



复位逻辑

若rst_n为0，count立即清零，确保复位信号优先级最高，模块可迅速回到已知状态。



计数与清零

当count等于count_max时，下一周期count归零，其余情况count自增1，实现循环计数。

完整代码展示

代码展示

```
module Tdcount
#(
    parameter count_max = 4'b1111  //计数最大值
)
(
input wire clk,
input wire rst_n,
output reg [3:0] count
);
always@(posedge clk or negedge rst_n) begin
if(!rst_n) begin //复位
count <= 0;
end
else if(count == count_max) begin //计数到达最大值
count <= 0;
end
else begin //计数
count <= count+1;
end
end
```



PART

04

ModelSim仿真流程

PART

新建仿真工程

01

启动与新建

启动ModelSim，选择File>New>Project，指定工程目录与名称。

02

添加源文件

在弹出的对话框中点击Add Existing File，浏览并添加Tdcount.v源文件。

03

新建测试文件

再点击Create New File，新建tb_TDcount.v测试平台文件，语言选择Verilog。完成后工程树将同时列出源文件与测试文件。

测试平台编写

时间定义

测试平台使用`timescale 1ns/1ps定义时间单位与精度。内部声明reg型clk与rst_n用于激励，wire型count接收输出。

信号激励

initial块先将rst_n置0维持20ns完成复位，随后置1释放；always #10 clk=~clk产生周期20ns的时钟。通过实例化Tdcount并传递参数count_max=2'd3，缩短仿真周期以便观察。



仿真代码完整



完整代码

```
`timescale 1ns/1ns
module tb_TDcount();
reg clk;
reg rst_n;
wire [3:0] count;
initial begin //初始化
    clk = 1;
        rst_n <= 0;
        #20
        rst_n <= 1;
    end
    always #10 clk = ~clk; // 产生周期为20ns的时钟信号
// 实例化计数模块
```

```
TDcount
#(
    .count_max(2'd3) //计数最大值
)
TDcount_inst
(
    .clk(clk),
    .rst_n(rst_n),
    .count(count)
);
endmodule
```



PART

05

波形验证与分析

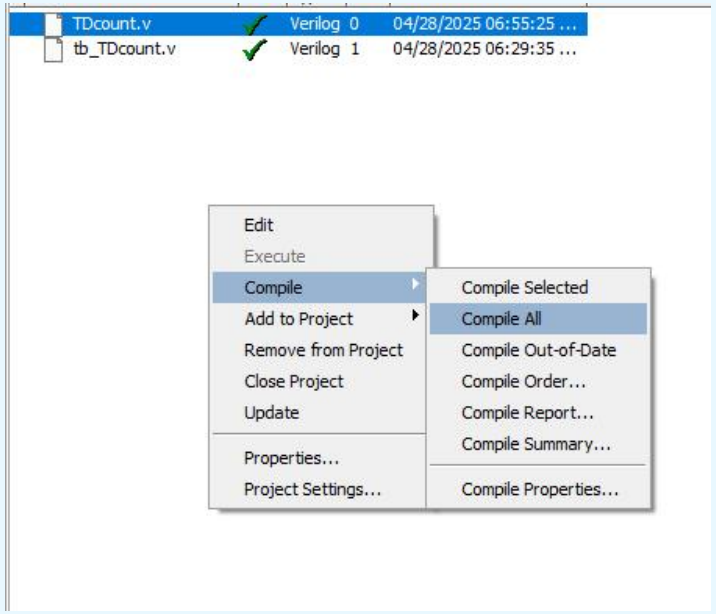
PART

编译与启动仿真

编译

在ModelSim工程空白处右键选择Compile>Compile All完成文件编译。

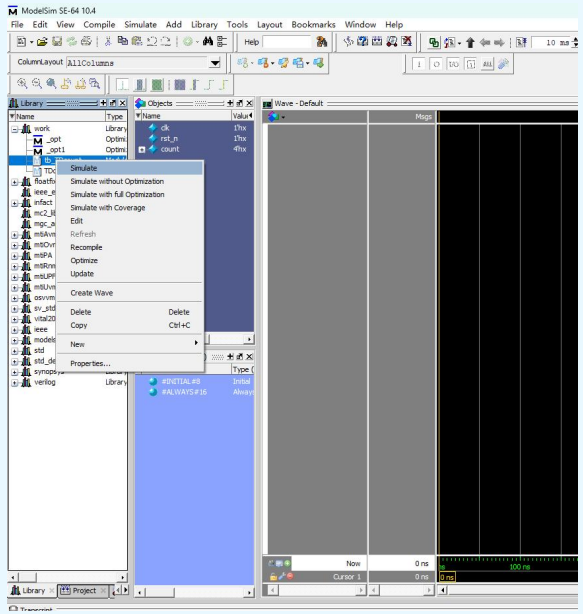
01



启动仿真

编译成功后展开Library窗口，定位到work.tb_TDcount，右键选择Simulate启动仿真。

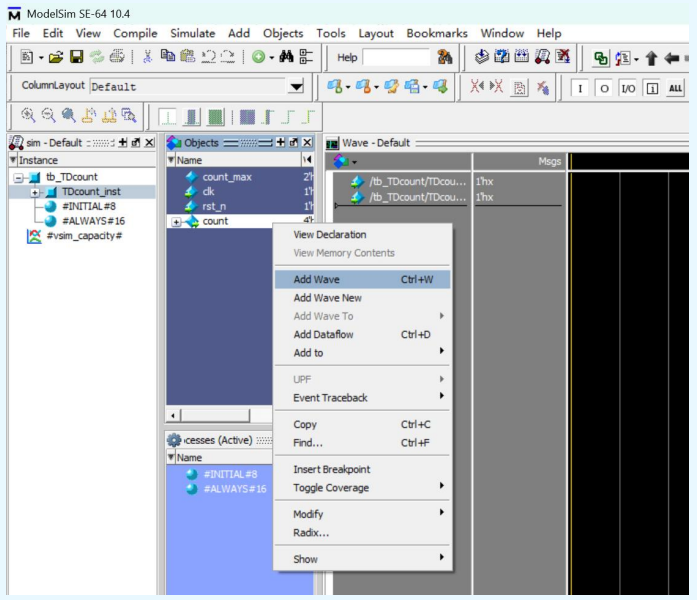
02



添加波形

随后右键count信号选择Add Wave加入波形窗口，设置仿真运行时间100ns并点击Run，即可开始观察信号变化。

03





波形结果解读



波形解读

仿真波形显示：复位期间count保持0；复位释放后，每个clk上升沿count递增，依次为0-1-2-3-0循环。时钟周期20ns，复位持续20ns，功能完全符合预期。通过缩短count_max至3，波形在100ns内即可展示4个完整计数周期，验证计数、清零、复位逻辑全部正确。

THANK YOU

感谢大家观看

