



中国科学技术大学  
University of Science and Technology of China

# 数字电路实验

## Lab2 测试与仿真

2024/10/8

# 实验介绍



中国科学技术大学  
University of Science and Technology of China

## ■ 在本次实验中，我们将要学习如何编写测试文件以及如何使用 Vivado 进行仿真。

一个测试工程师走进一家酒吧，要了一杯啤酒；  
一个测试工程师走进一家酒吧，要了 0.7 杯啤酒；  
一个测试工程师走进一家酒吧，要了 -1 杯啤酒；  
一个测试工程师蒙着眼睛，倒退着走进一家酒吧；  
一个测试工程师走进一家酒吧，什么也没要；  
一个测试工程师走进一家酒吧，要了一杯烫烫烫的银斤拷；  
一个测试工程师走进一家酒吧，要了 NaN 杯 Null；

.....

测试工程师们满意地离开了酒吧。  
然后一名顾客点了一份炒饭，酒吧炸了。



## ■ Testbench

中文释义：试验台；测试架；试样；试验工作台

在数字电路设计领域指用某种语言编写的程序或模块，用于在模拟过程中执行和验证硬件模型的功能正确性。

Testbench 由**不可综合**的 Verilog 代码组成，下图展示了一个典型架构。



其中：激励（Stimulus Block）是专门为待测模块生成的输入，输出校验（Output Checker）用于检查被测模块的输出是否符合预期，而被测模块（Design Under Test, DUT）是我们编写的 Verilog 模块。

## ■ 最基本的 Testbench 框架

```
module Module_tb ();  
  
    // 定义并产生激励信号  
    // .....  
  
    Test_module #(  
        // 参数接口  
    ) test (  
        // 待测模块端口  
    );  
endmodule
```

与正常设计时的 Verilog module 不同，用于测试的顶层模块**没有外界输入和输出**，这是因为 Testbench 模块应当是完全独立的，不受外部信号的干扰。

Testbench 中可以省略输出校验部分，转为人工检查。

## ■ Verilog 中的时序控制

### ■ 时延控制

基于时延的时序控制使用符号 **#**，它指定了语句从开始执行到执行完毕之间的时间间隔。时延可以是数字、标识符或者表达式。

### ■ 事件控制

在 Verilog 中，事件是指某一个 **reg** 或 **wire** 型变量发生了值的变化。基于事件触发的时序控制包括上升沿控制、下降沿控制、双边沿控制，使用符号 **@**。

## ■ 时序控制示例

```
wire A, B, X;  
reg Y;  
  
assign #10 X = A & B;  
always @(*) begin  
    #10;  
    Y = A & B;  
end  
// A&B 的计算结果延时 10  
// 个时间单位赋值给 X 和 Y
```

```
always @(clk) q <= d;  
// 信号 clk 只要发生变化, 就执行 q<=d  
  
always @(posedge clk) q <= d;  
// 信号 clk 上升沿时刻, 执行 q<=d  
  
always @(negedge clk) q <= d;  
// 信号 clk 下降沿时刻, 执行 q<=d
```

## ■ 生成一个仿真用的时钟信号

在 Lab1 中我们介绍过，initial 块中编写的任何代码都会在开始时执行，但仅执行一次，而 always 块则会循环执行内部的代码。

```
initial clk = 0;  
always begin  
    clk = 1;  
    #20;  
    clk = 0;  
    #20;  
end  
// 周期为 40 个时间单位的时钟
```

```
initial clk = 0;  
always #20 clk = ~clk;  
// 周期为 40 个时间单位的时钟
```

这里的 initial 语句不可缺省，否则 clk 信号没有初始值

## ■ 基本信息

实验软件: Vivado 2023.1、2019.1

流程: 创建项目→编写设计文件→编写仿真文件→运行仿真

```
module Counter #(
    parameter MAX_VALUE = 8'd100
)(
    input      clk,
    input      rst,
    output     out
);
reg [7:0] counter;
```

```
always @(posedge clk) begin
    if (rst)
        counter <= 0;
    else begin
        if (counter >= MAX_VALUE)
            counter <= 0;
        else
            counter <= counter + 8'b1;
        end
    end

    assign out = (counter == MAX_VALUE);
endmodule
```



# Vivado 仿真



中国科学技术大学  
University of Science and Technology of China

流程：创建项目→编写设计文件→编写仿真文件→运行仿真

The screenshot displays the Vivado IDE interface with several panels highlighted by colored boxes:

- Project Manager (Red box):** Located on the left, it contains a tree view with categories like Settings, IP Integrator, Simulation, RTL Analysis, Synthesis, Implementation, and Program and Debug. The 'Project Manager' category is highlighted in red.
- Sources (Green box):** Located in the center-left, it shows a list of sources including Design Sources, Constraints, Simulation Sources, and Utility Sources. The 'Sources' panel is highlighted in green.
- Project Summary (Blue box):** Located on the right, it provides an overview of the project settings, including Project name, Project location, Product family, Project part, Top module name, Target language, and Simulator language. The 'Project Summary' panel is highlighted in blue.
- Design Runs (Purple box):** Located at the bottom, it shows a table of design runs with columns for Name, Constraints, Status, WNS, TNS, WHS, THS, WBSS, TPWS, Total Power, Failed Routes, Methodology, RQA Score, GoR Suggestions, LUT, FF, BRAM, URAM, DSP, Start, Elapsed, and Run Strategy. The 'Design Runs' panel is highlighted in purple.

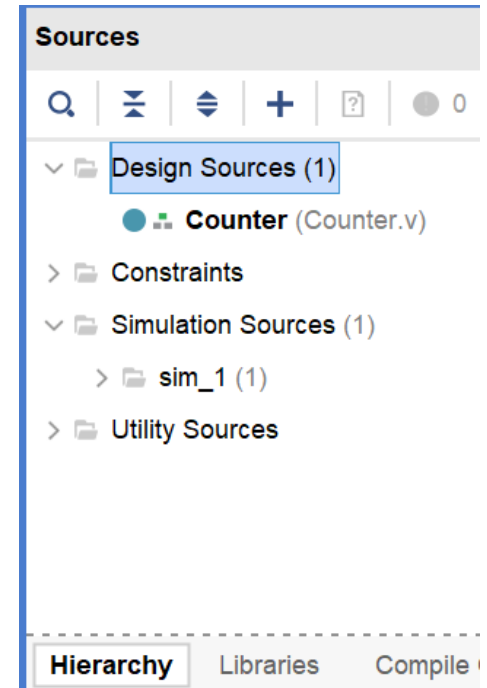
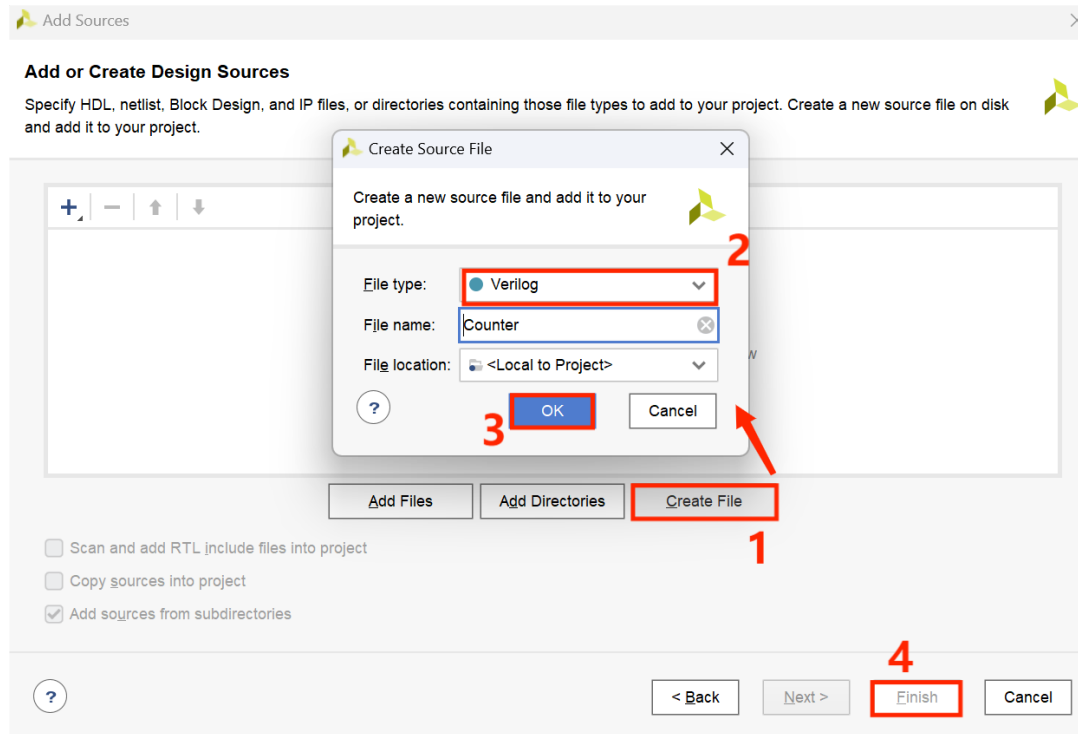
The 'Project Summary' panel shows the following details:

- Project name: Example
- Project location: C:/Users/19043/digital\_lab/Example
- Product family: Artix-7
- Project part: xc7a100tcsg324-1
- Top module name: Not defined
- Target language: Verilog
- Simulator language: Mixed

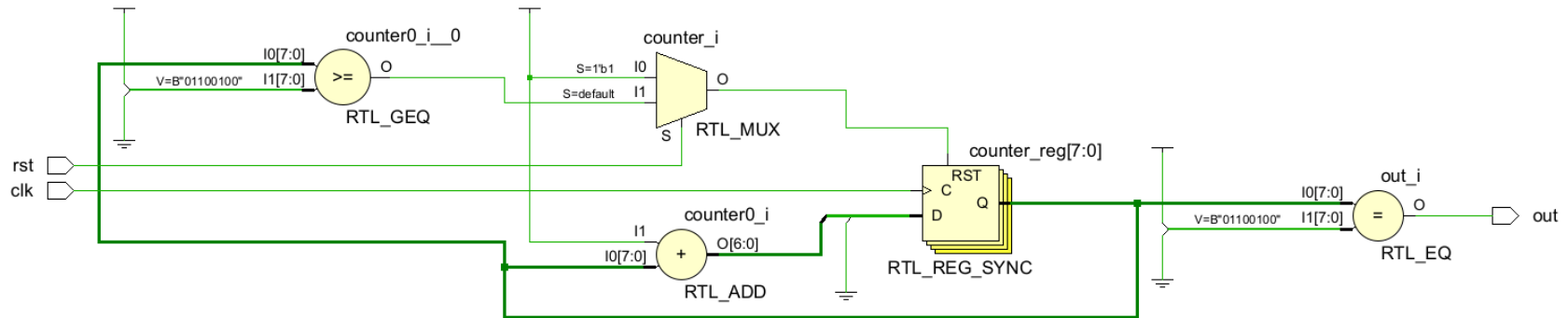
The 'Design Runs' panel shows the following table:

Name	Constraints	Status	WNS	TNS	WHS	THS	WBSS	TPWS	Total Power	Failed Routes	Methodology	RQA Score	GoR Suggestions	LUT	FF	BRAM	URAM	DSP	Start	Elapsed	Run Strategy
synth_1	constrs_1	Not started																			Vivado Synthesis Defa
impl_1	constrs_1	Not started																			Vivado Implementation

## ■ 在 Project Manager 窗口中点击 Add Sources, 选择 “Add or create design sources”



## ■ 点击左侧的 Open Elaborated Design , 画出设计电路图



```
always @(posedge clk) begin
  if (rst)
    counter <= 0;
  else begin
    if (counter >= MAX_VALUE)
      counter <= 0;
    else
      counter <= counter + 8'b1;
    end
  end
end
```

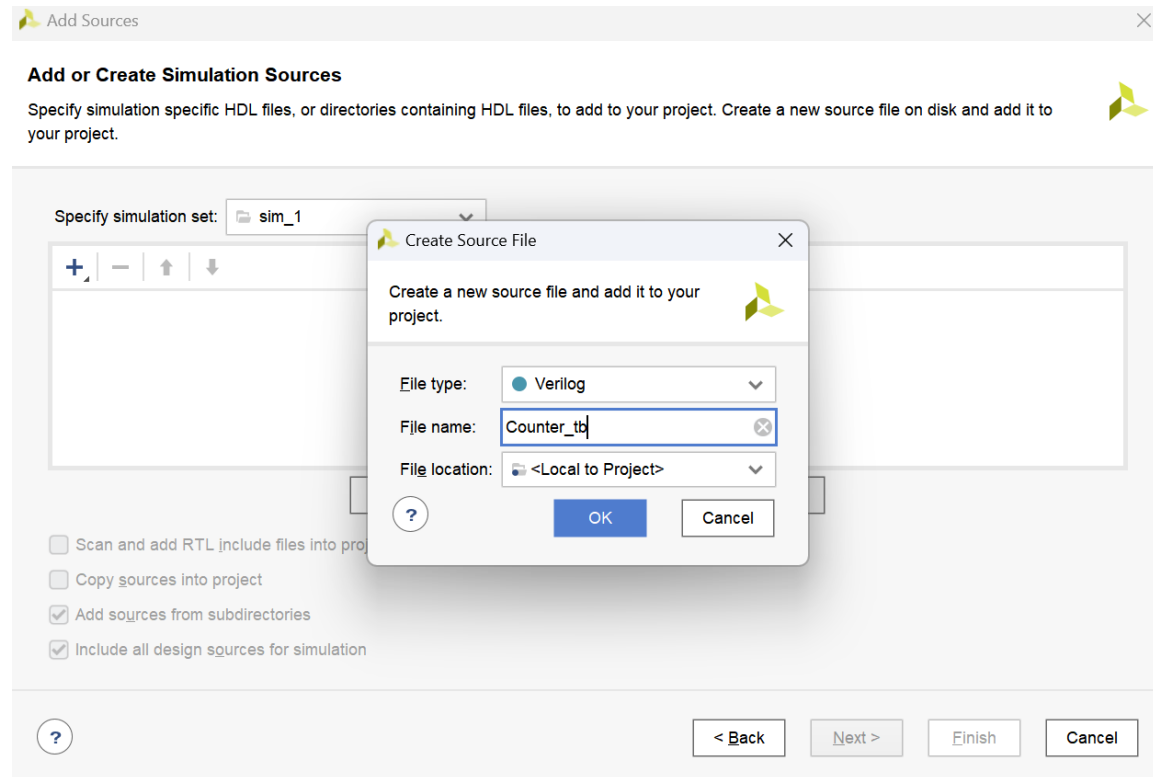
# Vivado 仿真



中国科学技术大学  
University of Science and Technology of China

## ■ 添加仿真模块 Counter\_tb

```
module Counter_tb();  
reg clk, rst;  
wire out;  
  
initial begin  
    clk = 0;  
    rst = 1;  
    #50;  
    rst = 0;  
end  
always #10 clk = ~clk;  
  
Counter #(8) my_counter (  
    .clk(clk),  
    .rst(rst),  
    .out(out)  
);  
endmodule
```



## ■ 点击左侧 SIMULATION 栏下的 Run Simulation 开始仿真



每经过 9 个时钟周期 out 就会发出一次信号，这是符合我们的设计预期的。

# 实验任务



中国科学技术大学  
University of Science and Technology of China

- [必做] 学习 Verilog 语言的进阶语法知识;
- [必做] 学习仿真文件的编写;
- [必做] 学习使用 Vivado 进行仿真;
- [必做] 按要求完成 Lab2 的实验练习题。

**实验检查 DDL: 10.17**

**报告提交 DDL: 10.24**

---

**谢谢！**