

# 第4章 基于ModelSim的仿真

---

- 4.1 ModelSim的仿真措施
- 4.2 Quartus II和ModelSim联合仿真

# 4.1 ModelSim的使用措施

---

## □4.1.1 ModelSim软件简介

ModelSim 是业界最优异的 HDL 语言仿真器。它提供最友好的调试环境，是业界唯一单一内核支持VHDL、Verilog HDL 和 SystemC 混合仿真的仿真器，同步也支持业界最广泛的原则如 Verilog 2023、SystemVerilog 等，内部集成了用于 C/C++，PLI/FLI和 SystemC 的集成 C 调试器。支持众多的 ASIC和 FPGA厂家库，能够用于 FPGA 和 ASIC 设计的 RTL 级和门级电路仿真。是作 FPGA/ASIC 设计的RTL 级和门级电路仿真的首选全方面支持UNIX(涉及 64 位)、Linux和 Windows平台。

---

---

ModelSim分几种不同的版本：SE、PE和OEM，其中集成在 Actel、Atmel、Altera、Xilinx以及Lattice等FPGA厂商设计工具中的均是其OEM版本。例如为Altera提供的OEM版本是ModelSim-Altera, 为Xilinx提供的版本为ModelSim XE. SE版本为最高级版本, 在功能和性能方面比OEM版本强诸多, 例如仿真速度方面, 还支持PC 、 UNIX 、 LIUNIX混合平台。

---

## 为何要学Modelsim?

1.

2.

## Modelsim的安装

---

---

同许多其他软件一样，Modelsim SE一样需要正当的License,一般我们用Kengen产生license.dat。

(1) 解压安装工具包开始安装，安装时选择Full product安装。当出现Install Hardware Security Key Driver时选择否。当出现Add Modelsim To Path选择是。出现Modelsim License Wizard时选择Close。

(2) 在C盘根目录新建一种文件夹flexlm，用Keygen产生一种license.dat,然后复制到该文件夹下。

(3) 修改系统的环境变量。右键点击桌面我的电脑图标，属性->高级->环境变量->(系统变量)新建。按下图所示内容填写，变量值内假如已经有别的途径了，请用“;”将其与要填的途径分开。LM\_LICENSE\_FILE = c:\flexlm\license.dat

# 仿真

---

仿真分为功能仿真, 门级仿真, 时序仿真

## 功能仿真(前仿真, 代码仿真)

主旨在于验证电路的功能是否符合设计要求, 其特点是不考虑电路门延迟与线延迟, 主要是验证电路与理想情况是否一致。可综合FPGA代码是用RTL级代码语言描述的, 其输入为RTL级代码与Testbench. 在设计的最初阶段发觉问题, 可节省大量的精力

## 门级仿真和时序列仿真 (后仿真)

使用综合软件综合后生成的门级网表进行仿真, 不加入时延文件的仿真就是门级仿真. 能够检验综合后的功能是否满足功能要求, 其速度比功能仿真要慢, 比时序仿真要快.

在门级仿真的基础上加入时延文件(.sdf)的仿真就是时序仿真, 比较真实地反应了逻辑的时延与功能. 综合考虑电路的途径延迟与门延迟的影响, 验证电路能否在一定时序条件下满足设计设想的过程, 是否存在时序违

- 
- 4.1.2 基本仿真环节
  - 建立工作库/建立资源库
    - 编译源代码
    - 开启仿真器
    - 执行仿真
-

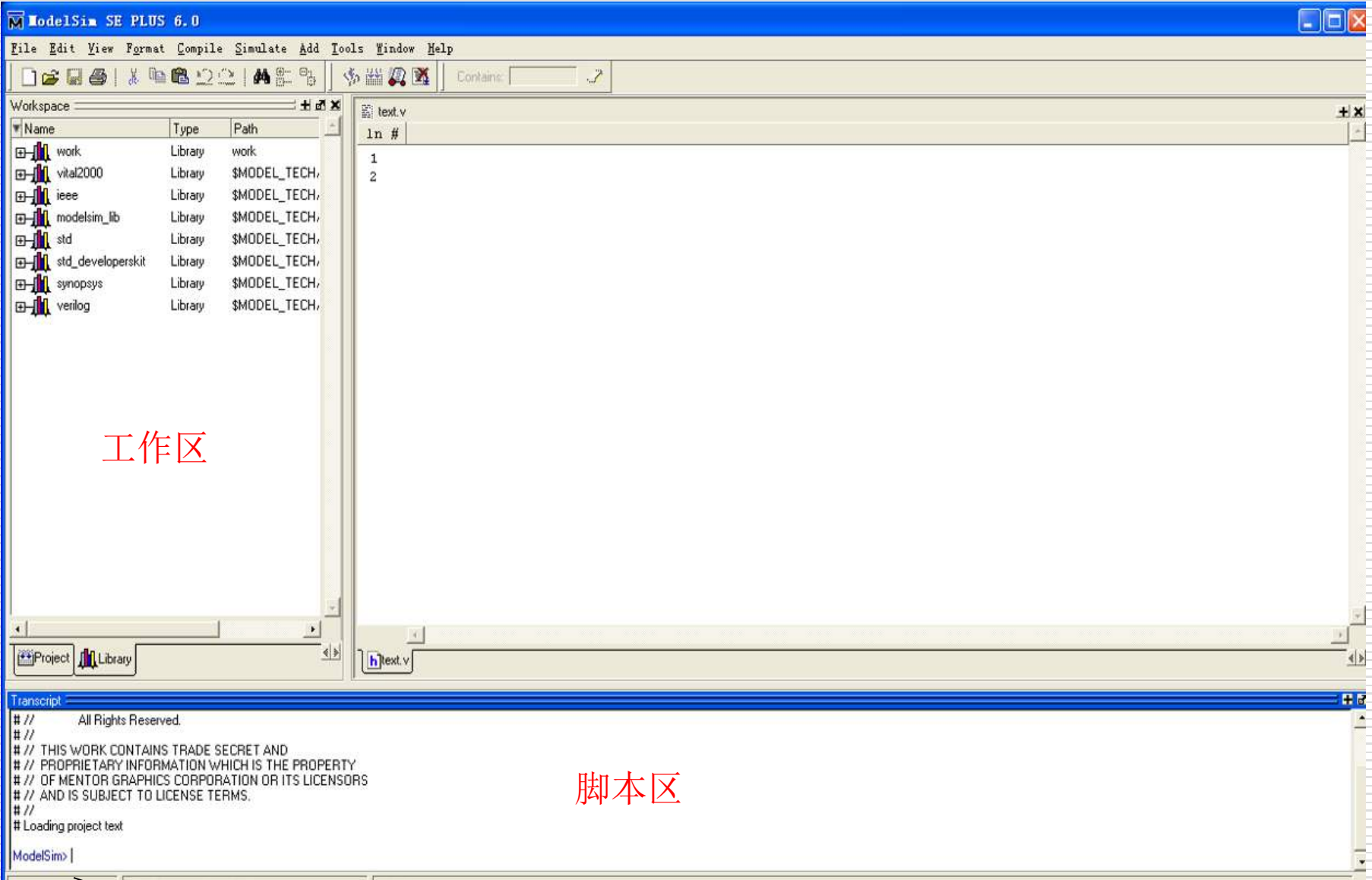
---

## □ 4.1.3 各个界面简介

- ModelSim仿真软件在默认条件下提供了主窗口、构造窗口、源程序窗口、信号窗口、进程窗口、变量窗口、数据流窗口、波形窗口、存储器窗口、列表窗口等 **11** 种不同的顾客窗口。



- 
- 主窗口
  - 主窗口在ModelSim开启时直接打开的，是全部其他窗口运营的基础。一般情况下主窗口分为工作区和脚本区（也叫命令控制台）两个部分，经过工作区能够很以便地对目前的工程的工作库以及全部打开的数据集合等进行控制，经过命令控制台能够在 ModelSim的提醒符下输入全部 ModelSim命令，而且可将命令执行成果反馈回来，便于实时掌握运营情况。主窗口的经典形式如所示。
-



工作区

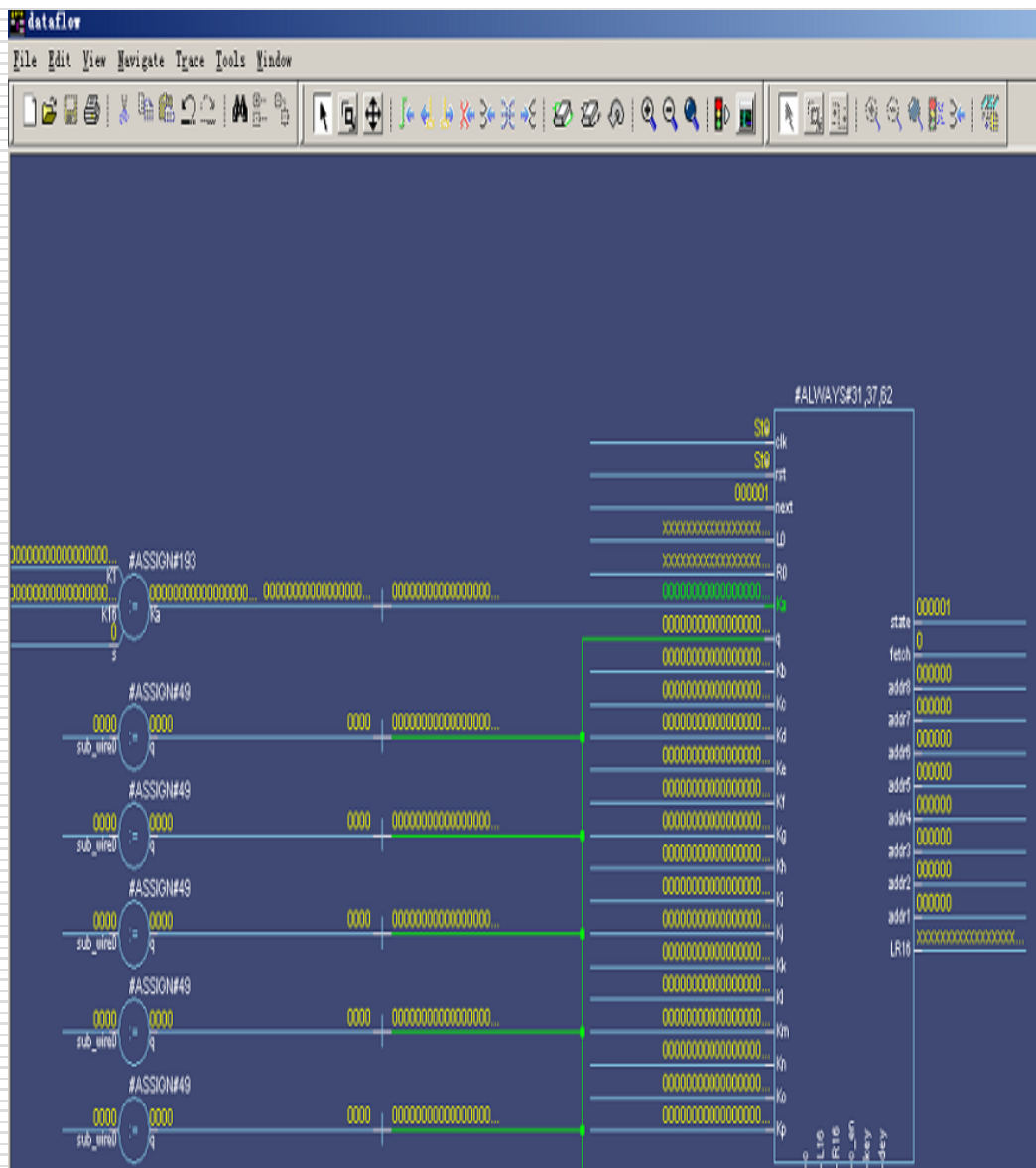
脚本区

命令输入

---

## ■ 数据流窗口

- 数据流窗口（**Dataflow**）是一般仿真软件都提供的一种通用窗口，经过该窗口能够跟踪设计中的物理连接，跟踪设计中事件的传播，也能够用来跟踪寄存器、网线和进程，极大地丰富了调试措施。数据流窗口中能够显示进程（能够是 **Verilog** 的一种模块）、信号、网线和寄存器等，也能够显示设计中的内部连接。窗口中有一种内置的符号表，映射了全部的 **Verilog** 基本门，例如与门、非门等，这些符号能够在数据流窗口中显示。其他的 **Verilog** 基本组件能够使用模块或者顾客定义的符号在数据流窗口中显示。
-

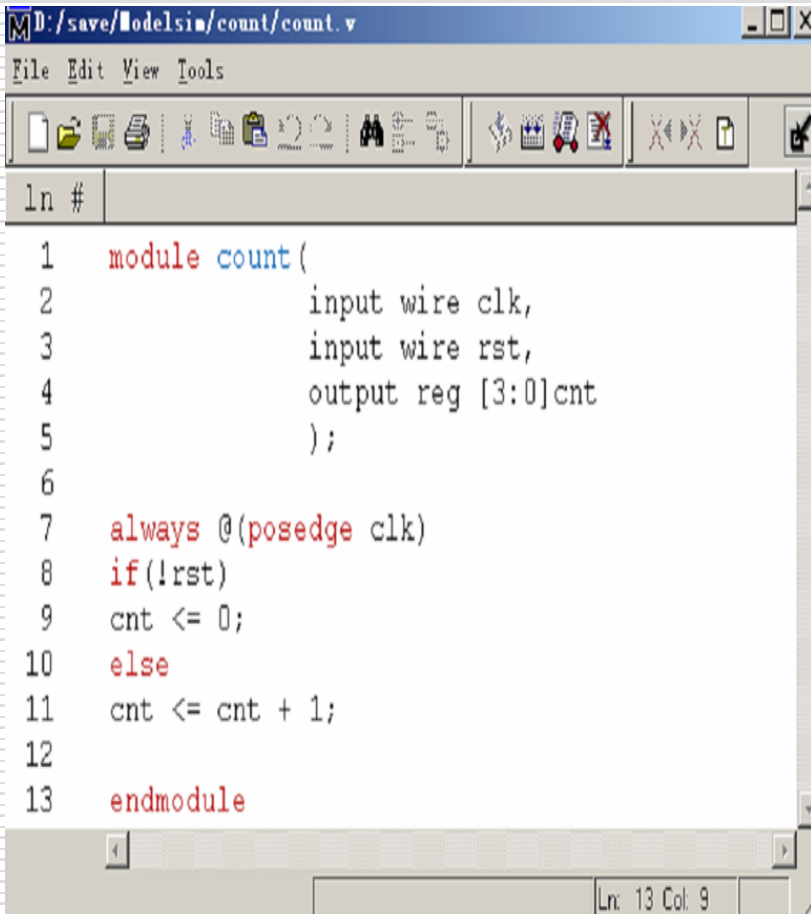


- 数据流窗口中的符号都使用了类似 `[#ASSIGN#23#2]`、`[#ALWAY#56]`或 `[<module_name>]`等信息进行阐明，其中第一种#阐明了这个符号的语句，第二个#背面紧跟了产生这个符号的语句所在的源文件中的行号，第三个#阐明了这个语句在源文件中属于目前行的第几种语句。

ns	delta	/test/u1/clk	/test/u1/rst	/test/u1/cnt
0	+0	StX	StX	xxxx
0	+1	St0	St0	xxxx
10	+1	St1	St0	0000
20	+1	St0	St0	0000
30	+1	St1	St0	0000
40	+1	St0	St0	0000
50	+1	St1	St0	0000
60	+1	St0	St0	0000
70	+1	St1	St0	0000
80	+1	St0	St0	0000

## ■ 列表窗口

- 列表窗口使用表格的形式显示仿真的成果。窗口被分为两个可调整的部分，右边为信号列表，左边为仿真时间以及仿真的 **Delta** 时间。同步能够从主窗口中创建列表窗口的第二个副本，两个列表窗口能够进行不同的设置，便于仿真成果的比较，同步也能够对波形比较时对相应得数据进行列表对比。



The screenshot shows the ModelSim source file editor window. The title bar indicates the file path is 'M:\save\ModelSim\count\count.v'. The menu bar includes 'File', 'Edit', 'View', and 'Tools'. The toolbar contains various icons for file operations and editing. The main text area displays the following Verilog code:

```
ln #
1  module count(
2      input wire clk,
3      input wire rst,
4      output reg [3:0]cnt
5  );
6
7  always @(posedge clk)
8  if(!rst)
9  cnt <= 0;
10 else
11 cnt <= cnt + 1;
12
13 endmodule
```

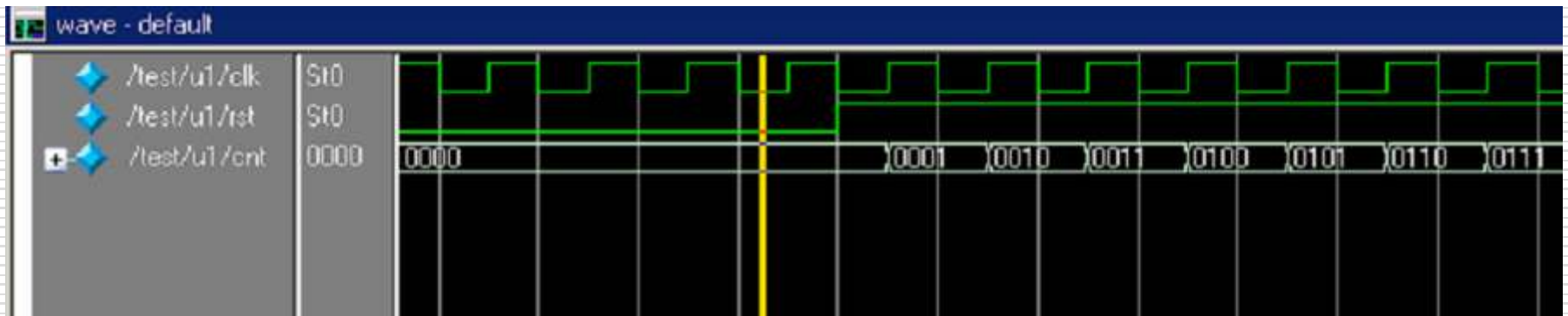
The status bar at the bottom right shows 'Ln: 13 Col: 9'.

## 源文件窗口

- 源文件窗口主要用来显示和编辑 HDL 源文件代码。  
ModelSim 源文件窗口是一种很优异的硬件描述语言编辑工具，在这个窗口中能够显示文件的行号，同步能够使用“新建”按钮打开语言模板来以便源代码的编写，语言模板会根据编写的源文件的类型自动调整。

## ■ 波形窗口

- 就像列表窗口能够用来查看仿真成果一样，波形窗口也能够用来显示仿真波形，而且比列表窗口更直观，所以波形窗口是最常用的仿真窗口之一。波形窗口一般分为 3 个不同区域，分别用来显示信号名称以及途径，光标所在位置信号的目前值、波形等。



---

## □ 4.1.4 ModelSim调试功能

- 在波形窗口中监视信号，查看仿真的波形
    - 变化目前目录到要变异的文件所在的目录
    - 建立工作库并编译有关源文件
    - 加载设计的鼓励文件
    - 向波形窗口中添加项目，即要监视的信号
    - 缩放波形显示
    - 在波形窗口中使用光标
    - 保存波形窗口格式
-



---

## □ 4.1.4 ModelSim调试功能

### ■ 使用断点、断点设置措施

- 在源文件窗口中打开要调试的源文件，找到要设置断点的地方
  - 在要设置断点的行的红色行号背面单击鼠标左键，行号后出现的红色圆圈表达断点设置成功。
  - 当仿真被断点停止后能够查看信号目前值
    - 在信号窗口中查看各个信号的目前值
    - 在源程序窗口中，将光标悬停在需要查看的信号上，会出现一种注释阐明信号目前值
    - 能够经过主窗口命令台中使用**examine**查看
-

# 4.1.5 功能仿真

## 功能仿真需要的文件

1. 设计HDL源代码：能够使VHDL语言或Verilog语言。
2. 测试鼓励代码：根据设计要求输入/输出的鼓励程序
3. 仿真模型/库：根据设计内调用的器件供给商提供的模块而定，如：FIFO、ADD\_SUB等

## 仿真环节

### 以2选1多路器为例给出详细环节

#### 1. 开启modelsim软件

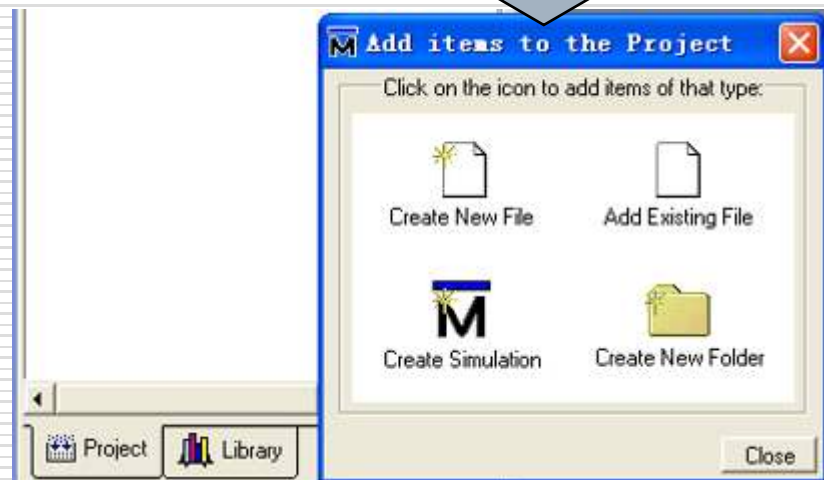
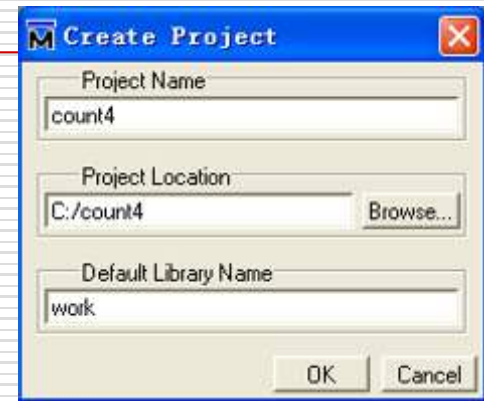
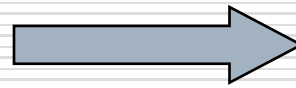
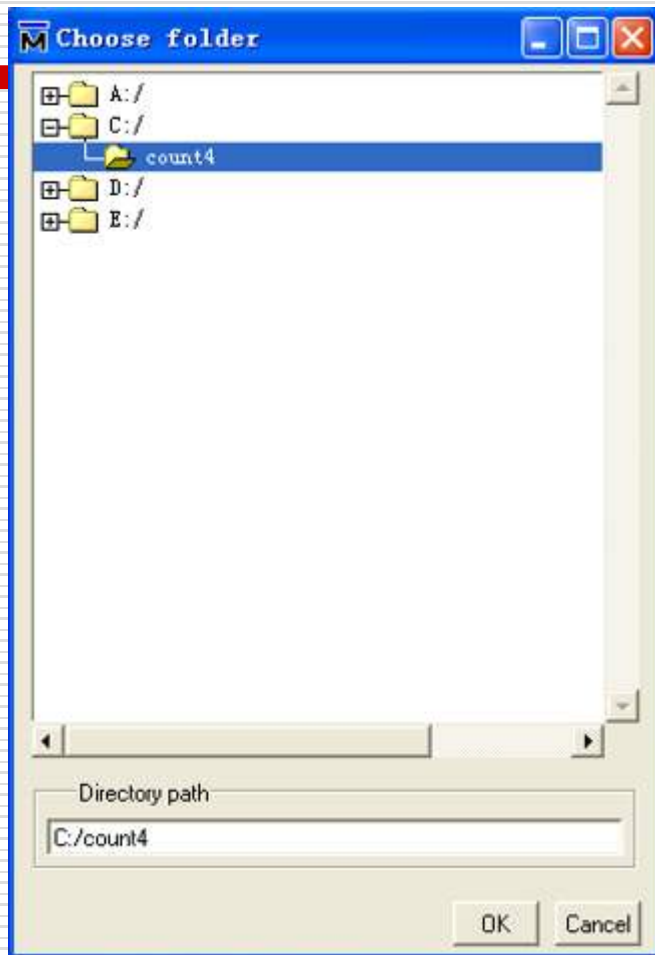
先在c盘建立文件夹count4, 在modelsim中选择File -> Change Directory, 在弹出的Choose folder对话框中设置目录途径为c:/XXXX

#### 2. 建立工程

在modelsim中建立project, 选择File -> New -> Project.

在Project Name栏中填写你的项目名字, 提议和你的顶层文件名字一致。Project Location是你的工作目录, 你可经过Brose按钮来选择或变化。Ddfault Library Name能够采用工具默认的work。

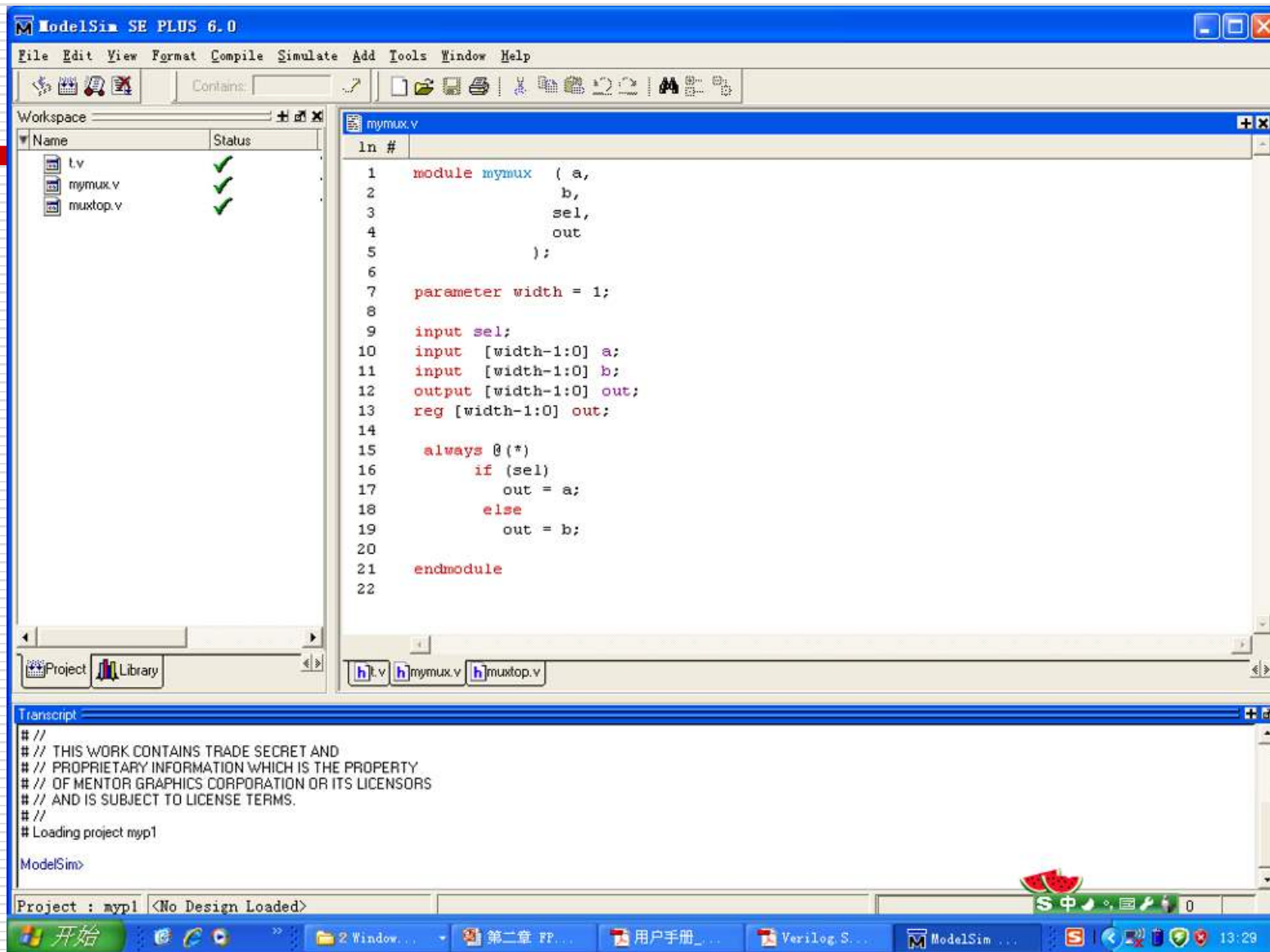
Workspace窗口的library中就会出现work库。



---

### 3. 为工程添加文件

工程建立后, 选择Add Existing File后, 根据相应提醒将文件加到该Project中



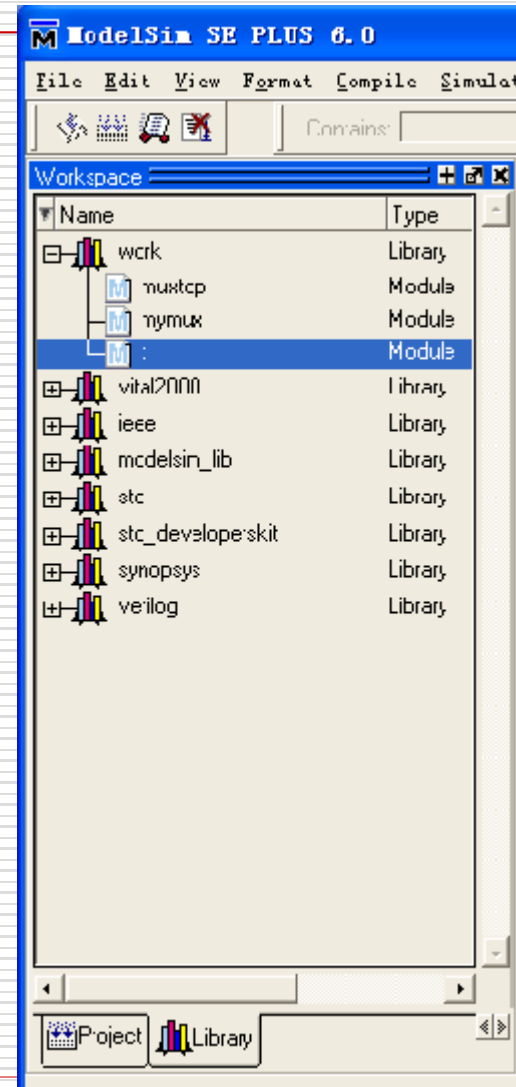
## 4.编译文件

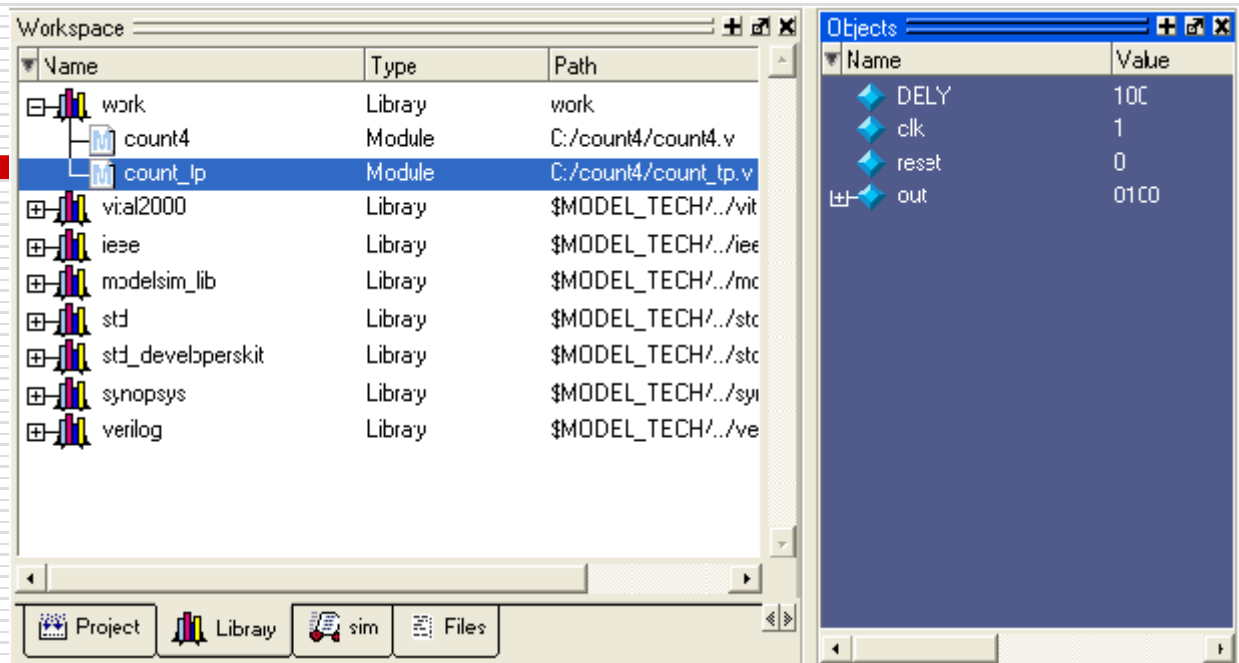
编译（涉及源代码和库文件的编译）。编译可点击Comlile⇒Comlile A11来完毕。



## 5.装载文件

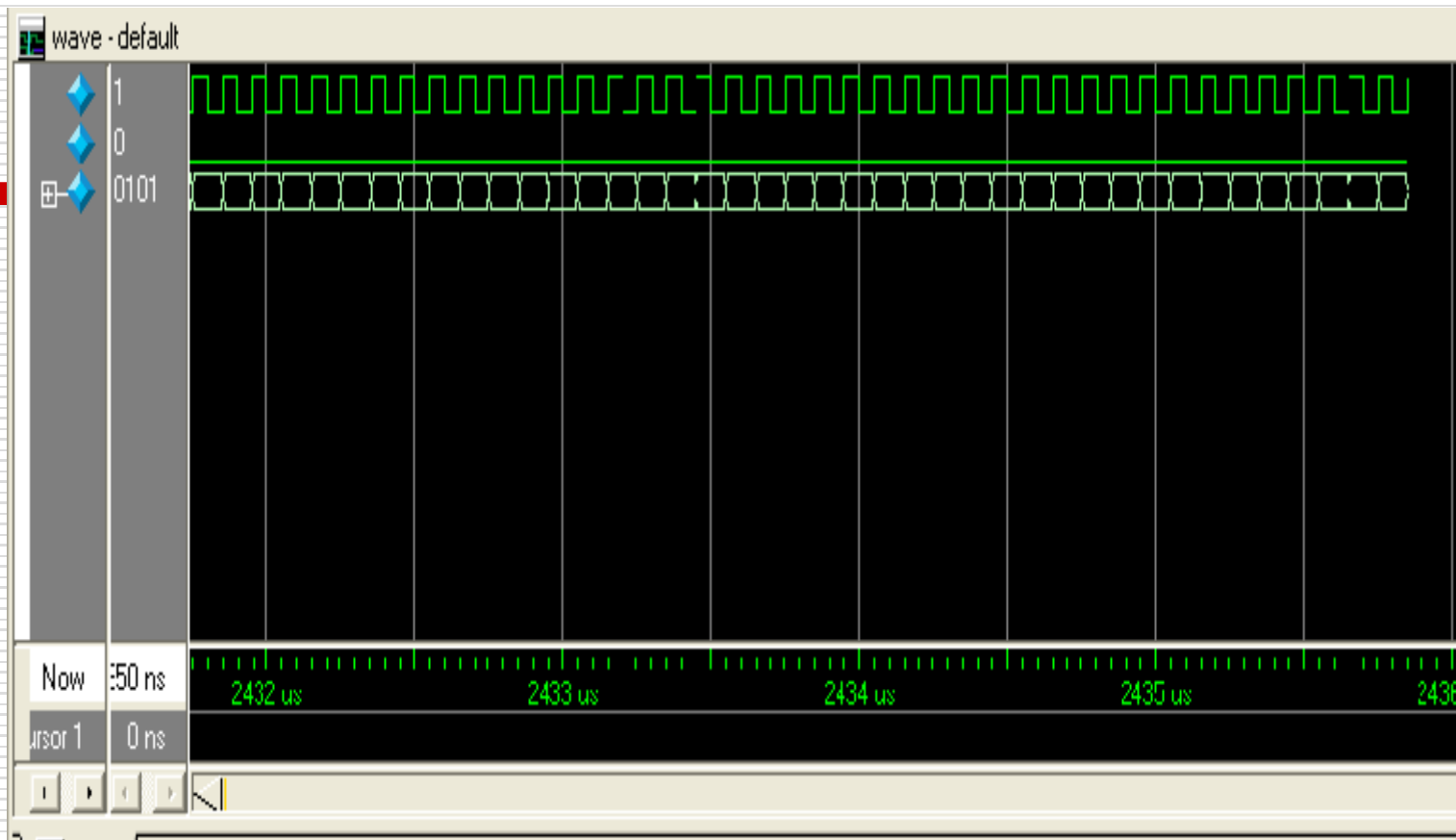
- (1)双击libray 中work中的t装载
- (2)点击simulate ⇒start simulation  
按右图设置,点击ok





## 6. 开始仿真

点击workspace下的sim, 点击count\_tp, 选择add → add to wave  
然后点run -all,开始仿真



## 7. 退出仿真

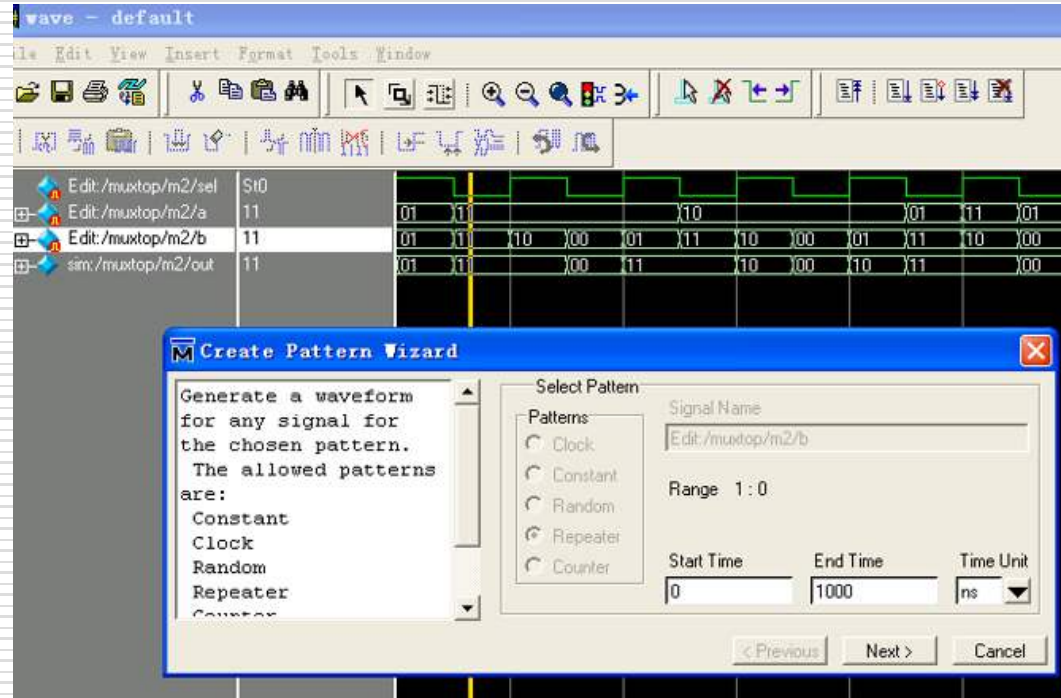
在仿真调试完毕后退出仿真, 在主窗口中选择simulate → end simulation



# 补充:

---

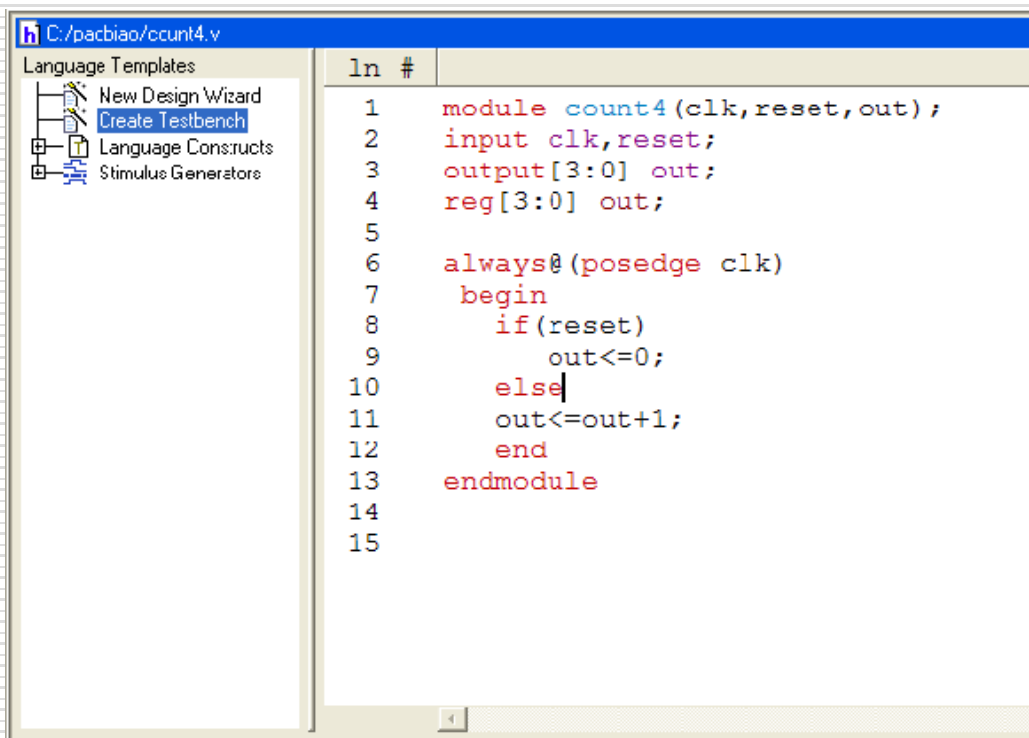
(1) 也能够不加testbench, 仿真环节跟前面相同, 装载文件时双击muxtop  
在sim中点击muxtop, add → add to wave  
对输入信号sel、a、b编辑测试波形



点击run -all

(2)我们能够在modelsim内直接编写Testbench

Modelsim提供了诸多Testbench模板，我们直接拿过来用能够降低工作量。点View->Source->Show Language Templates然后会出现一个加载工程，接着你会发觉在刚刚的文档编辑窗口左边出现了一个Language Templates窗口



The screenshot shows the ModelSim interface with the Language Templates window open. The window title is "C:/pacbiao/ccunt4.v". The left pane shows a tree view with "Language Templates" expanded, and "Create Testbench" selected. The main pane displays a Verilog code template for a 4-bit counter, with line numbers 1 through 15 on the left.

```
ln #
1  module count4(clk,reset,out) ;
2  input clk,reset;
3  output[3:0] out;
4  reg[3:0] out;
5
6  always@(posedge clk)
7  begin
8      if(reset)
9          out<=0;
10     else
11         out<=out+1;
12     end
13 endmodule
14
15
```

---

展开Verilog项，双击Creat Testbench会出现一种创建向导选择Specify Design Unit工作库下的目的文件，点Next能够指定Testbench的名称以及要编译到的库等，此处我们使用默认设置直接点Finish。这时在Testbench内会出现对目的文件的各个端口的定义还有调用函数

接下来，设计者能够自己往Testbench内添加内容了，然后保存为.v格式即可。按照前面的措施把Testbench文件也编译到工作库中。

---

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/967041025010006164>