

Spring, 2020



ModelSim®

1090140071

FPGA设计及应用

ModelSim软件操作简介

大连理工大学 电信学部
夏书峰

ModelSim仿真软件



- Model**Sim**是MentorGraphics公司开发的行业内最通用的FPGA仿真器之一, 着重编译和仿真工作

- 支持Verilog HDL、VHDL、SystemVerilog、SystemC等主流HDL语言的编译和仿真

- ModelSim的部分产品

<http://www.model.com/>

- ◆ **ModelSim OEM**

给Altera的版本是ModelSim AE, 给Xilinx的版本是ModelSim XE

- ◆ **ModelSim ModelSim PE (Personal Edition)**

基础功能版本, 可混合仿真Verilog 和 VHDL

- ◆ **ModelSim ModelSim DE (Deluxe Edition)**

PSL & System Verilog assertions, Code Coverage, Enhanced Dataflow, Waveform Compare, and support for Xilinx SecureIP as standard

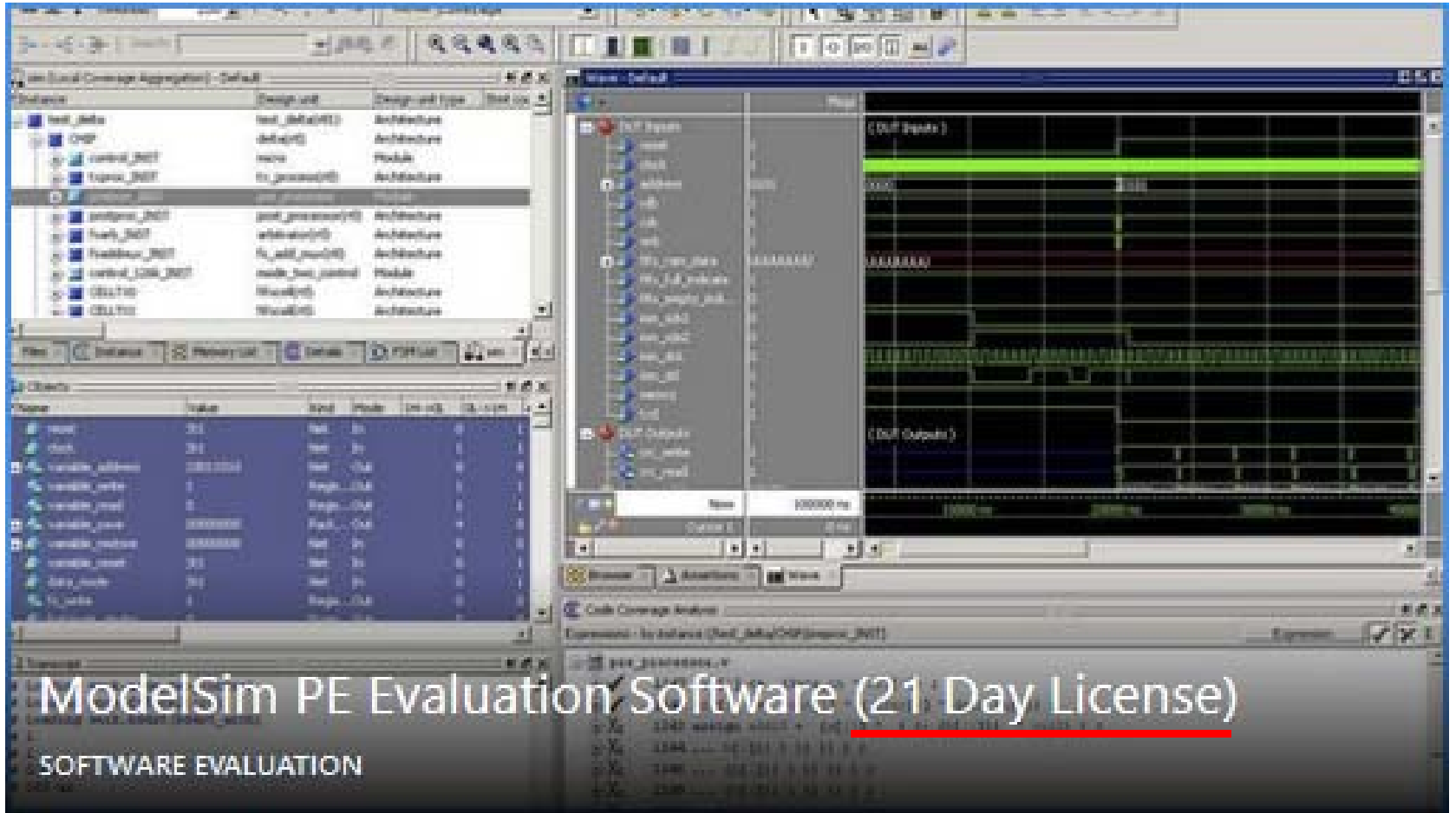
- ◆ **ModelSim ModelSim SE (System Edition)**

适合仿真非常大的设计, 高性能高容量, 支持代码覆盖和debug, ASIC gate-level sign-off

ModelSim SE与OEM版本的区别

- | | ModelSim SE | ModelSim XE II |
|-----------------|-------------|----------------|
| • 技术差异 | | |
| 单一语言支持 | ✓ | ✓ |
| 混合语言支持 | ✓ | — |
| Xilinx库支持 | ✓ | ✓ |
| 其它库支持 | ✓ | — |
| • 调试功能差异 | | |
| ModelSim SE | | ModelSim XE II |
| 基本图形用户界面 | ✓ | ✓ |
| SignalSpy | ✓ | — |
| 代码覆盖率检查 | ✓ | — |
| 性能分析 | ✓ | — |
| 数据流与X跟踪 | ✓ | — |
| 波形比较 | ✓ | — |
| 检查点复原 | ✓ | — |
| 混合语言调试 | ✓ | — |
| Debug Detective | ✓ | — |
- 以ModelSim XE版本为例，对于代码少于40000行的设计，ModelSim SE比XE要快近10倍。对于代码超过40000行的设计，SE要比XE快近40倍。

从Model网站可以申请PE版21天试用许可证

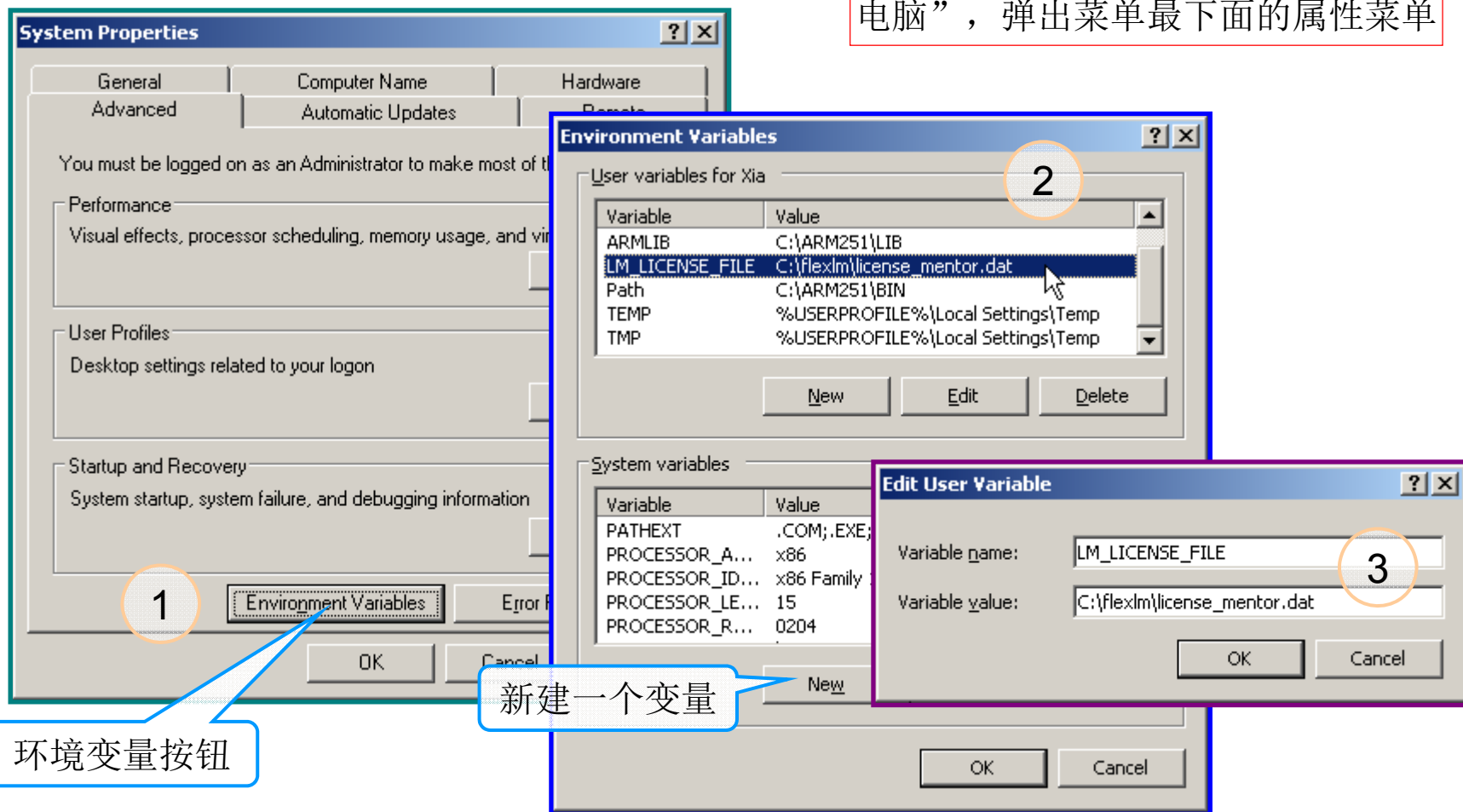


<http://www.model.com/> 或
<https://www.mentor.com/products/fpga/model/>

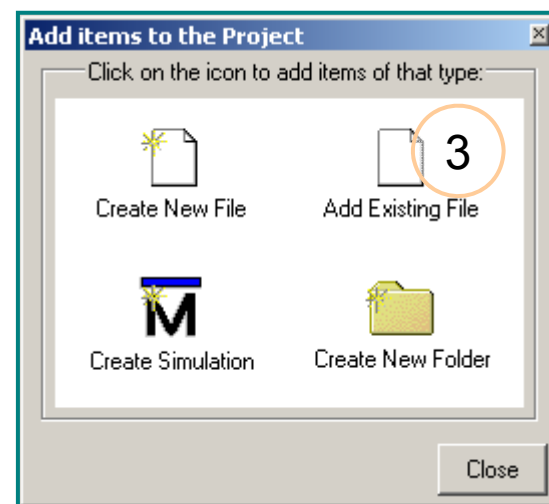
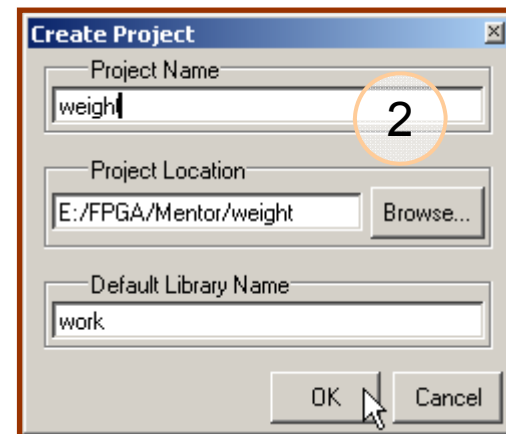
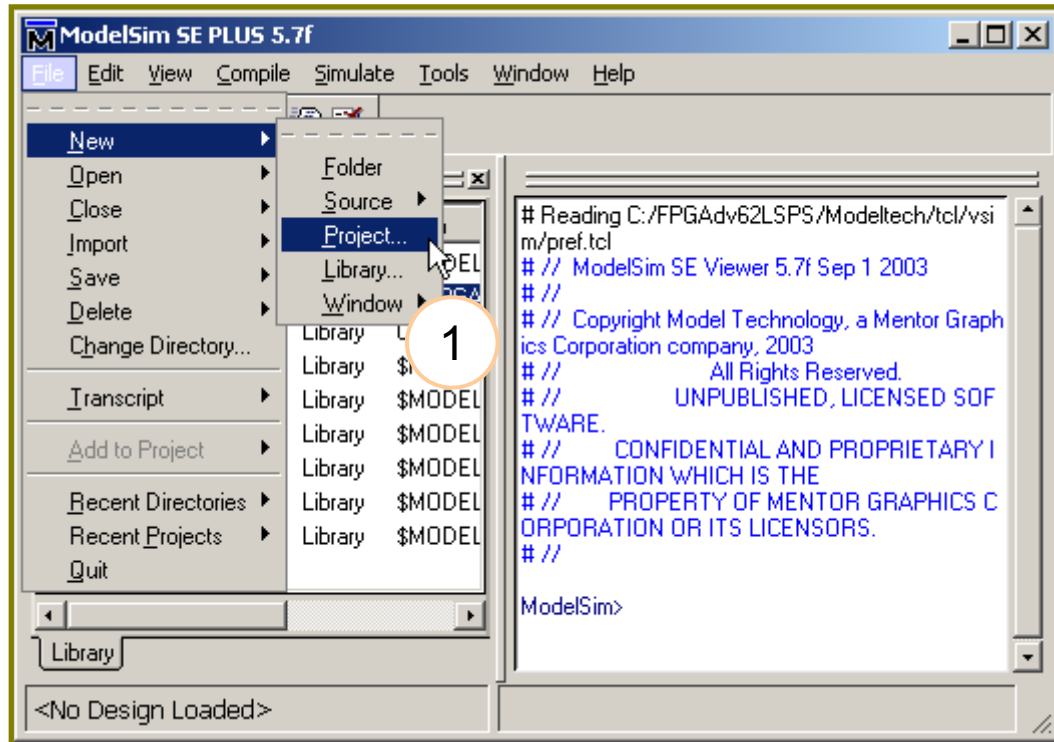
用系统环境变量指定License文件

手工添加MGLS_LICENSE_FILE或LM_LICENSE_FILE向系统环境变量，Mentor软件启动后会自动寻找该变量指向的license文件。

系统属性对话框用鼠标右键击“我的电脑”，弹出菜单最下面的属性菜单



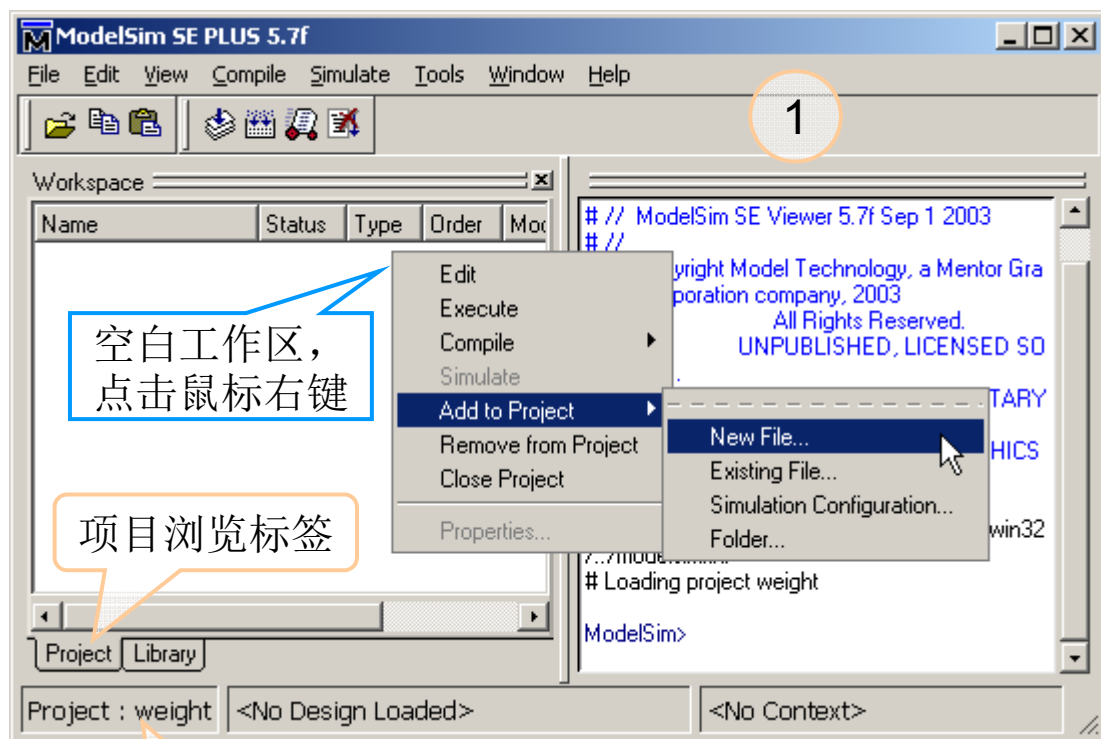
新建工程



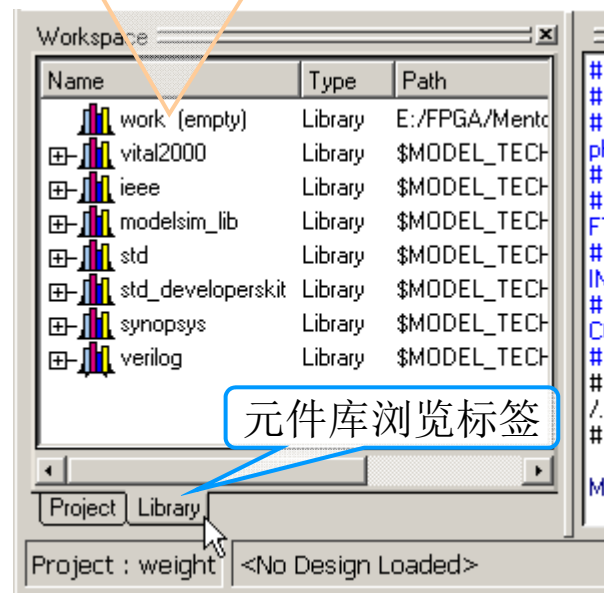
ModelSim的工程管理基于工程(Project),

- 启动图1所示界面后在主菜单File—New—Project...新建一个空项目
- 弹出图2所示的对话框，在第一行输入工程名，并在第二行指定一个目录单独存放设计文件，第三行是当前工作库，默认work，不必改变
- 在图3所示的对话框里可以新建文件，也可以添加现有的文件，建立新目录是为了在项目视图里方便项目的层次管理，并不在磁盘上物理的建立目录。这个对话框可以先关闭，下一步用鼠标右键菜单可以执行同样功能。

新建设计文件



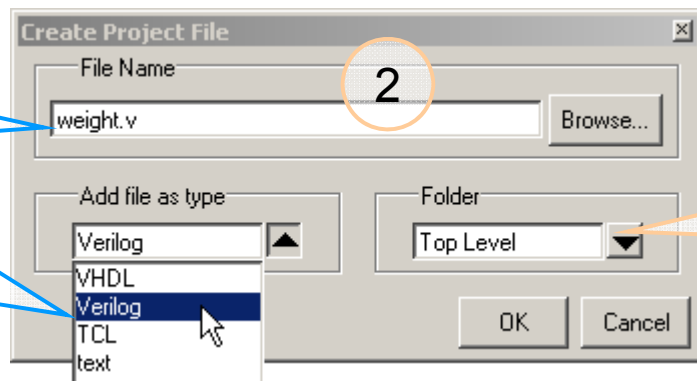
- 项目建立完毕, 在工作库标签里已经出现空的work工作库
- 若没有出现刚才默认的work库, 可以关闭工程再重新打开
- 或从File>ChangeDirectory... 菜单找到刚建立的weight文件夹(里面有个work目录, 但不要进到work目录里面去)



工程名称

新建文件名

选择新建文件的类型



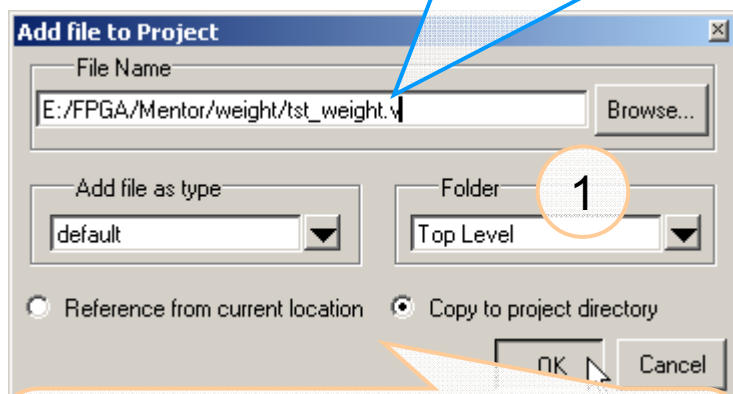
选择新建文件的目录层次, TopLevel是Workspace的根目录, 若此前在workspace里建了子目录, 可在列表里选择

添加已有的设计文件

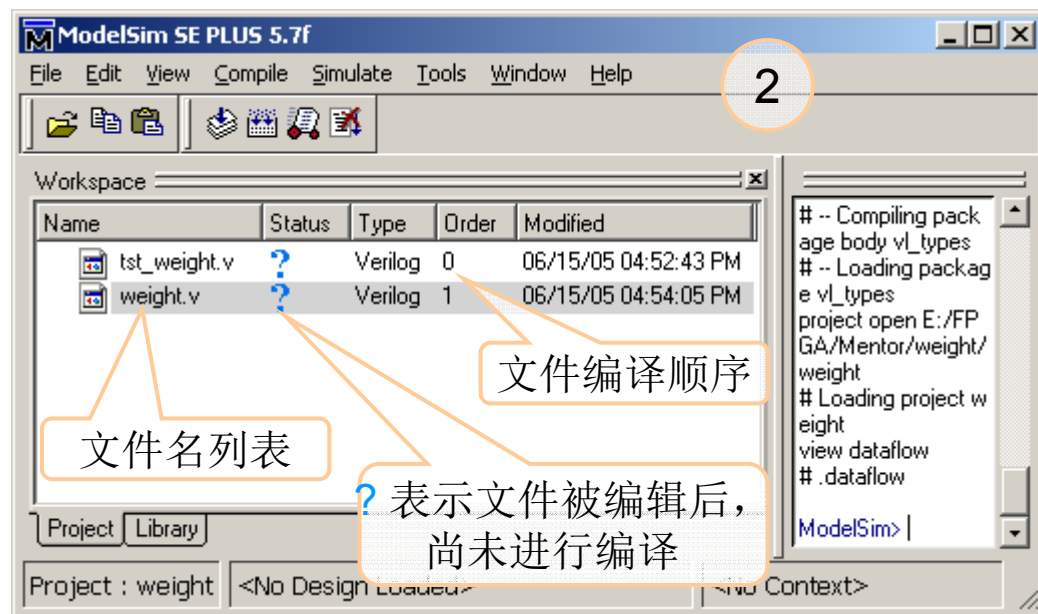
除了如前所述新建程序文件外，还可以向工程中添加已有的设计文件，可以通过如下两种方式弹出添加文件对话框：

- 在前面建立工程后弹出的那个“Add Items to the Project”对话框中按“Add Existing File”按钮
- 在空白工作区击鼠标右键，弹出菜单选Add To Project → Existing File...

浏览找到文件,可一次选择多个添加

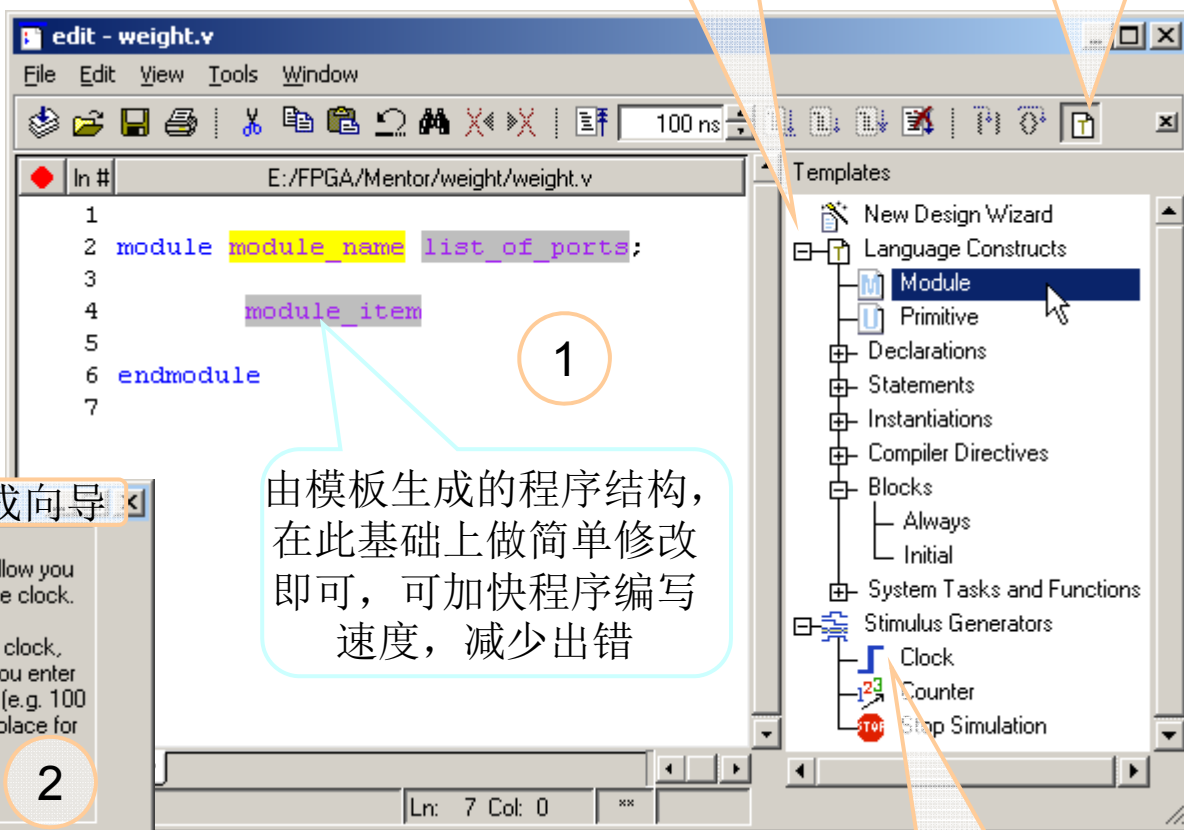


Reference from current location是从文件原来位置引用文件，Copy to project directory是将设计文件拷贝到当前工程的目录。为避免引用的文件被改动，建议copy文件到工程目录。

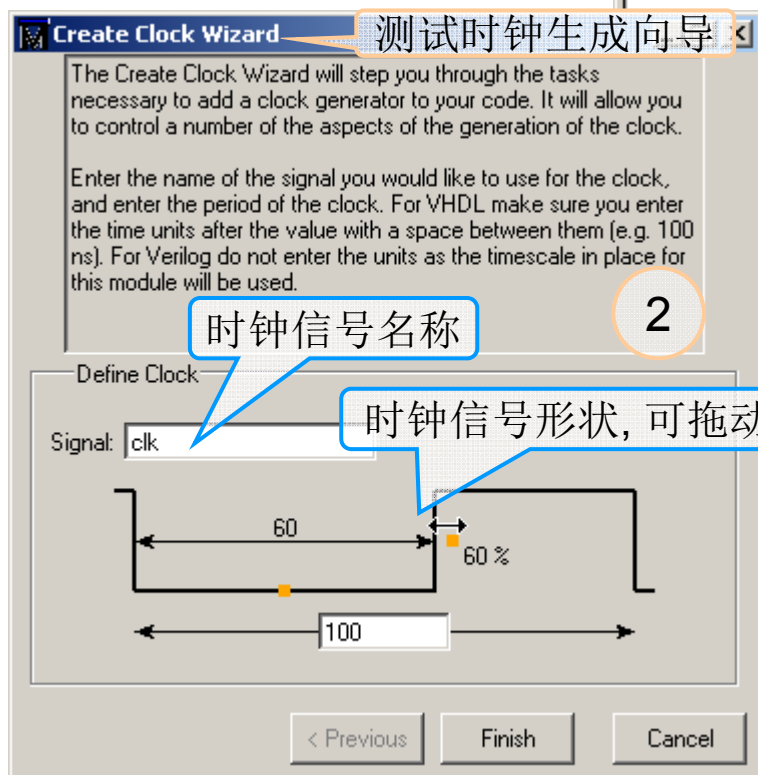


编写程序

双击workspace里的文件名称将弹出支持语法着色的代码编辑器，带有右图所示的语言模板。双击语言模板里的Module项，就会在当前光标处添加一个module-endmodule结构，方便程序编写。此外还有许多模板，如生成时钟的模板



由模板生成的程序结构，在此基础上做简单修改即可，可加快程序编写速度，减少出错



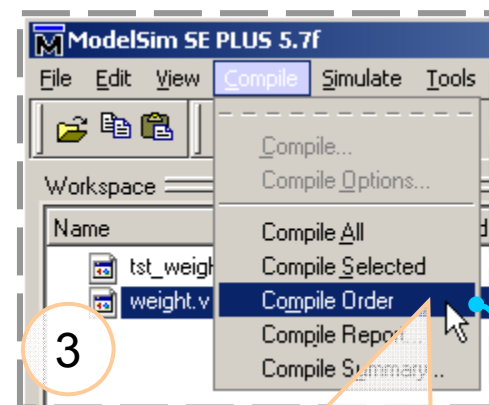
```
5 reg clk;
6 initial begin
7     clk = 0;
8     forever begin
9         clk = #30 1;
10        clk = #20 0;
11    end
12 end
13
```

生成测试信号的模板:如时钟和计数器模板

时钟模板生成的测试时钟代码段

录入代码、编译

- 从多个地方都可以编译当前的代码文件



编译错误定位

双击这条红色的错误信息将弹出图3代码编辑窗口并定位错误所在行

```
# Compile of weight.v failed with 1 errors.
# Compile of weight.v was successful.
# vlog_capture -work work E:/FPGA/Mentor/weight/weight.v
# vlog_capture -work work E:/FPGA/Mentor/weight/tst_weight.v
# Compile of tst_weight.v was successful.
# vlog_capture -work work E:/FPGA/Mentor/weight/weight.v
# vlog_capture -work work E:/FPGA/Mentor/weight/weight.v
# Compile of weight.v failed.
# There were 1 file(s) that could not be compiled successfully
# Compile of weight.v failed with 1 errors.
# Compile of tst_weight.v was successful.
# 2 compiles, 1 failed with 1 error.

ModelSim>
```

这里出现一条错误，双击这条红色的错误信息将弹出图2所示的对话框，说明具体错误情况

```
...or/weight/weight.v -- Unsuccessful Compile
vlog -work work E:/FPGA/Mentor/weight/weight.v
Model Technology ModelSim SE vlog 5.7i Compiler 2003.09 Sep 15 2003
-- Compiling module weight
** Error: E:/FPGA/Mentor/weight/weight.v(1): near "EOF": expecting IDENT
```

错误改正完毕并重新编译通过，就可准备下一步进行仿真工作

```
edit - weight.v
File Edit View Tools Window
100 ns
E:/FPGA/Mentor/weight/weight.v
1 module weight(a, b, c, r, g);
2   input a, b, c;
3   output r, g;
4   reg r, g;
5
6   always @ (a or b or c)
7   begin
8     case ({a,b,c})
9       3'b000,
10      3'b001,
11      3'b010: {r, g}=2'b00;
12      3'b011,
13      3'b100: {r, g}=2'b10;
14      3'b101,
15      3'b110,
16      3'b111: {r, g}=2'b11;
17      default: {r, g}=2'b00;
18    endcase
19  end
20 endmodule
21
```

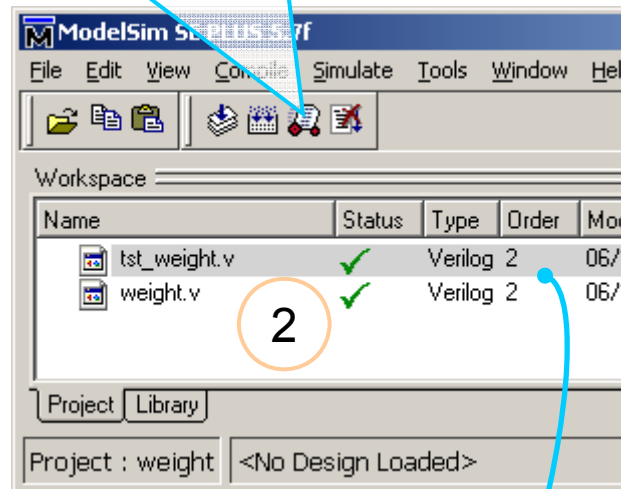
橙色反显有问题的程序行

实际是因为这里endmodule写错了

仿真和测试

- 一个被动模块若没有输入信号则不会对仿真过程产生变化的响应，因此若要测试一个模块，需要提供输入信号
- ModelSim没有提供像Quartus II的波形编辑器那样的仿真波形生成工具，因此需要手工编写测试向量（这是主流）
- ModelSim的测试信号可用Verilog语言编写，也可在vsim提示符下输入“force clock 1 50, 0 100 -repeat 100”这样的命令控制某个信号在某时刻状态，还可以在signal窗口选中信号，然后Edit-> Clock...分别设定各条信号的变化行为。
- 本例中weight.v是被测模块，而tst_weight.v则是生成测试输入信号a b c的test bench，如图1所示。

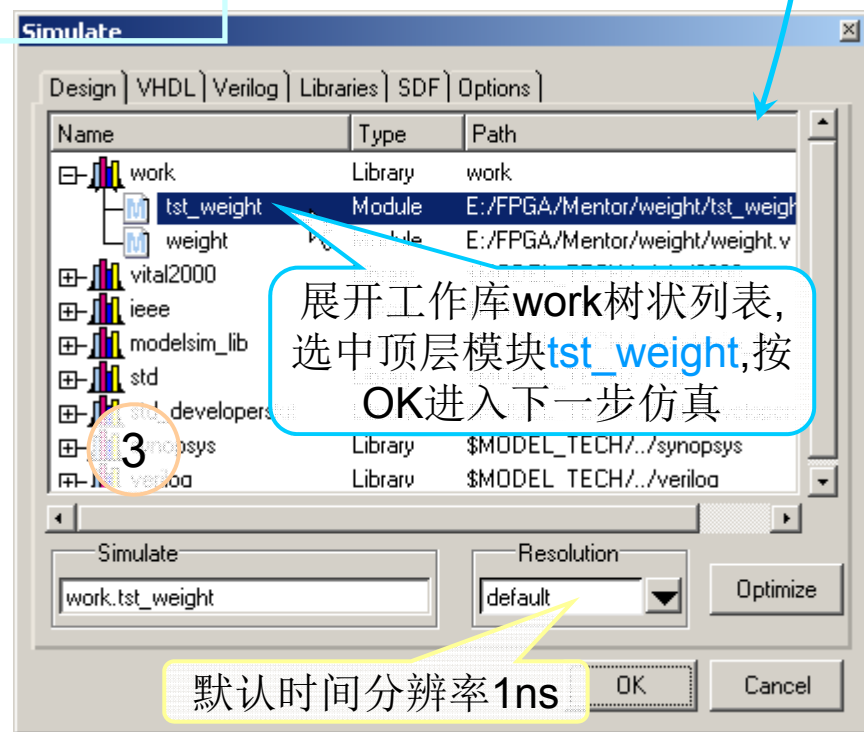
Simulate仿真按钮,也可从主菜单Simulate菜单启动仿真



```
edit - tst_weight.v
File Edit View Tools Window

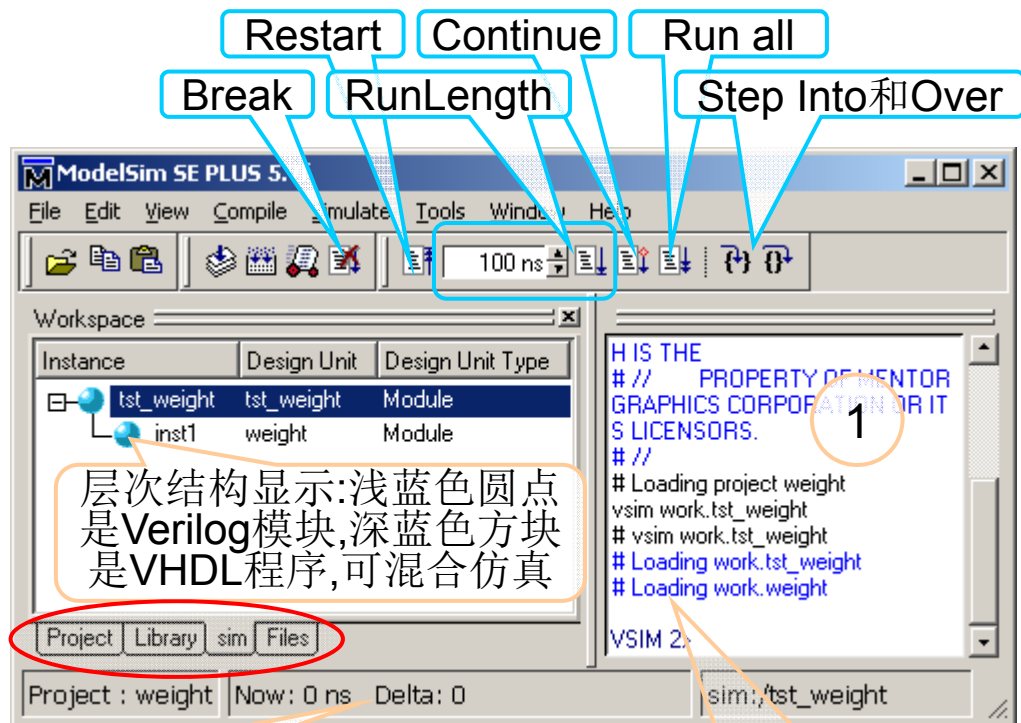
16 // generate test signal
17 always @(posedge clock) begin
18     {ain, bin, cin} = $random % 8;
19 end
20
21 weight inst1(.a(ain), .b(bin), .c(cin), .g(gou)
22
23 //clock generator
24 initial begin clock = 1'b0;
25     forever #5 clock = !clock;
26 end
27
28 endmodule
```

测试向量的代码窗口



默认时间分辨率1ns

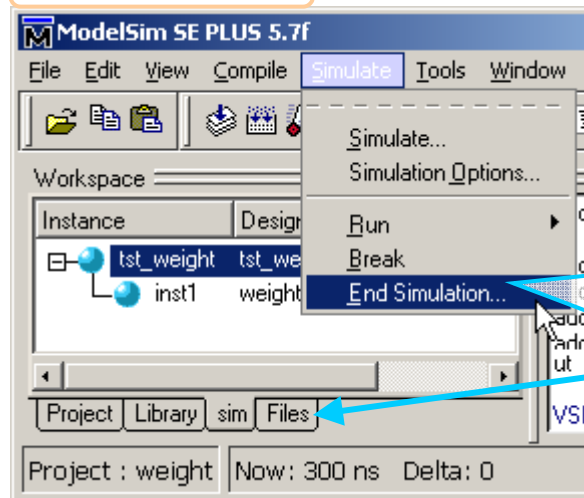
仿真主窗口和按钮功能



仿真状态信息

程序正确加载信息

停止仿真过程,工作区下面的sim和Files标签消失,只剩Project和Library标签



这些按钮在主菜单Simulate->Run的子菜单中都有对应项:

- **Restart:** 重新加载程序并重新运行代码,相当于复位。当仿真中源程序被改动后,要重新编译,编译完后按该按钮可自动重载程序并重新开始运行;
- **RunLength:** 运行指定时间长度,可由前面的微调框设定,默认每次运行100nS;
- **Continue:** 被Break按钮停止后可按此按钮再次开始继续运行;
- **Run all:** 连续运行程序,直到按下Break按钮或者遇到\$stop或\$finish指令才停下;
- **Step Into和Step Over:** 进入或者运行跳过某个函数等等。
- **Break:** 用于中断正在运行的程序回到命令交互界面。

ModelSim的众多窗口

ModelSim主控窗口

过程窗口, 这里列出了全部的过程块

变量窗口, 列出了当前过程的变量

只读的代码调试窗口可指示当前运行位置和断点等信息

View->All可以显示全部仿真窗口

信号窗口, 列出模块所有信号和端口

List窗口, 列出指定信号在仿真过程中随时间变化的情况, 滚动显示时上面的5行标题保持不动

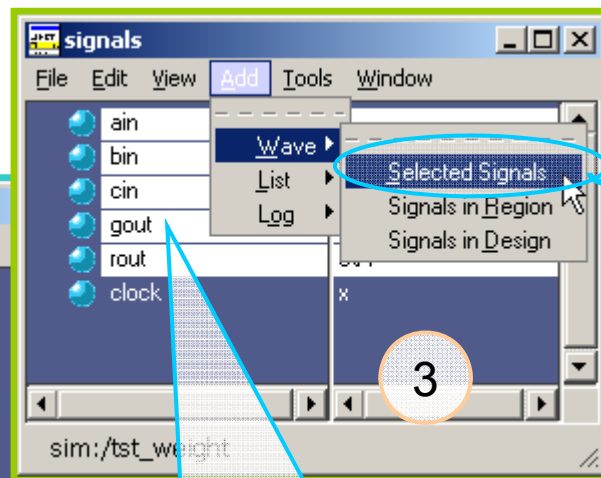
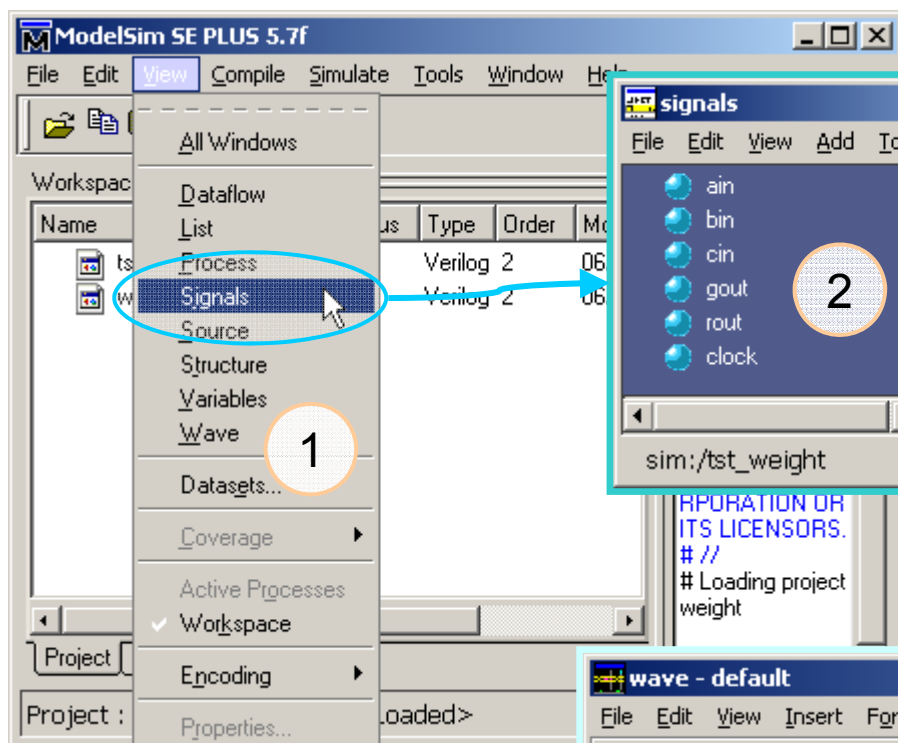
数据流窗口, 显示指定信号的流经途径

波形(wave)显示窗口, 显示指定的若干个信号的波形

The screenshot displays the ModelSim SE PLUS 5.7f interface with several windows open:

- Workspace:** Shows the project hierarchy with 'tst_weight' and 'inst1'.
- process:** Lists process blocks: '<Ready> #INITIAL#14', '<Done> #ALWAYS#17', and '<Ready> #INITIAL#24'.
- variables:** Lists variables for 'tst_weight': 'ain' (1), 'bin' (0), 'cin' (1), and 'clock' (1).
- source - tst_weight.v:** Shows the Verilog code for the 'tst_weight' module, including a clock generator.
- signals:** Lists signals and ports: 'ain' (1), 'bin' (0), 'cin' (1), 'gout' (St1), 'rout' (St1), and 'clock' (1).
- list:** Shows a table of signal values over time (ns, delta, and signal values).
- dataflow:** Shows the dataflow for the selected signal, with a message: 'Welcome to the Enhanced Dataflow Window! NOTE: Extended dataflow mode active'.
- wave:** Displays the waveform for the selected signals, showing digital logic levels over time (0 ns to 346 ns).

添加要测试的信号



用鼠标左键与Ctrl或Shift键组合可以在此选中需要观察的多个信号, 然后加入到Wave或List窗口等



仿真过程控制按钮

可弹出Dataflow窗口查看选中信号的驱动源及波形, 如左图5

加入信号自动弹出波形窗口

The screenshot shows the ModelSim SE PLUS 5.7f software interface. The 'Simulate' menu is open, and the 'Run All' option is highlighted with a blue selection bar. A red circle is drawn around the 'Run All' option. The status bar at the bottom shows 'Project : weight' and 'Now: 300 ns Delta: 0'.

“全速”运行后source窗口弹出,程序当前停在14行

可用这3个按钮放大或缩小波形
以方便查看波形细节或全景

实心红块表示有效的断点

空心红块表示Disabled
(被禁用)的断点

可在行前用鼠标右键菜单添加, 删除, 编辑或禁用断点

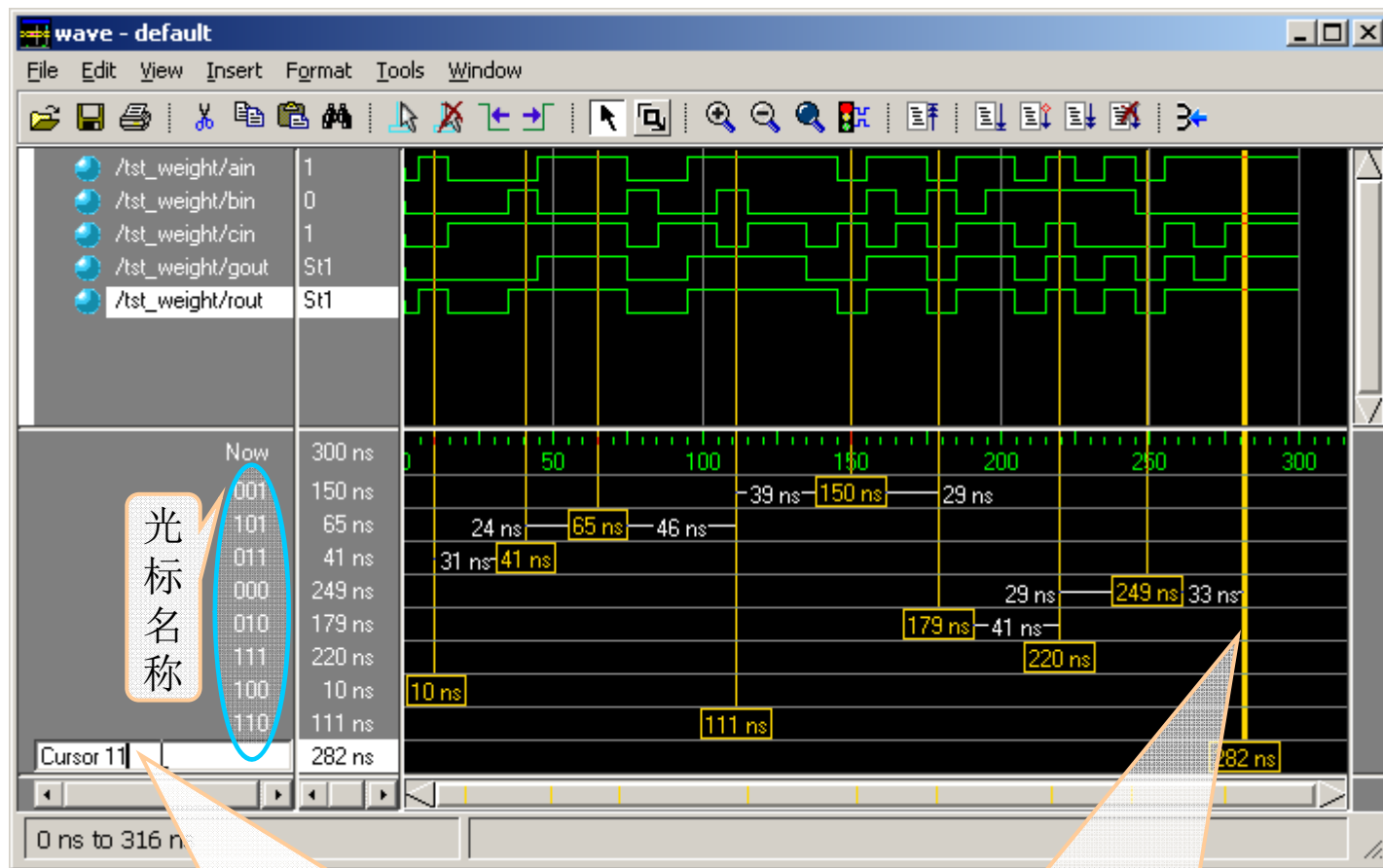
寻找跳变发生位置

添加或删除时间标记

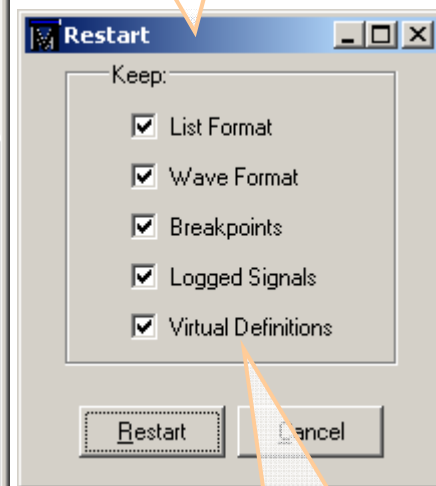
当前时间标记处各信号取值,可用鼠标右键设定不同显示数制

时间标记(可拖动)

观察、验证仿真结果



仿真过程若需要
改动程序则改动
完毕并编译通过
后还要**restart**才
能刷新仿真数据,
弹出如下对话框



可以添加并移动多个时标,
也可用鼠标右键给光标改名

粗线这个是当前时标, 信号取值窗
口显示的是当前时标处信号值

Restart时这个对话框里的选项选中

- 以上是行为仿真, 没有引入延时等信息, 还看不到以前提到的竞争和冒险等现象; 延时信息的仿真可以通过综合器生成的*.sdf文件以“反向标注”等方式实现。