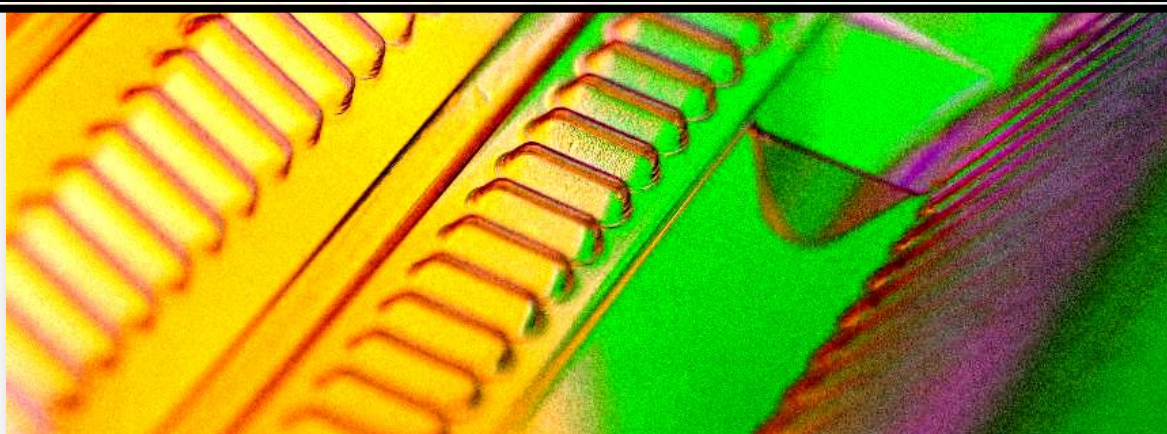


zrtech

**FPGA/CPLD 开发套件实验教程**

**--仿真，调试，设计篇**



**WWW.ZR-TECH.COM**

## 实验二、modelsim 仿真指南（一）

### 实验目的：

本节给大家介绍一种采用 modelsim 进行 FPGA 设计仿真的方法，使用户掌握采用 modelsim 进行三种不同层次的仿真

### 实验原理：

#### 1. FPGA设计仿真的三个层次

##### 纯功能仿真

在完成一个设计的代码编写工作之后，可以直接对代码进行仿真，此时成为纯功能仿真，检测源代码是否符合功能要求。这时，仿真的对象为 HDL 代码，比较直观，速度比较快，可以进行与软件相类似的多种手段的调试（如单步执行等）。在设计的最初阶段发现问题，可以节省大量的精力。

##### 综合后仿真

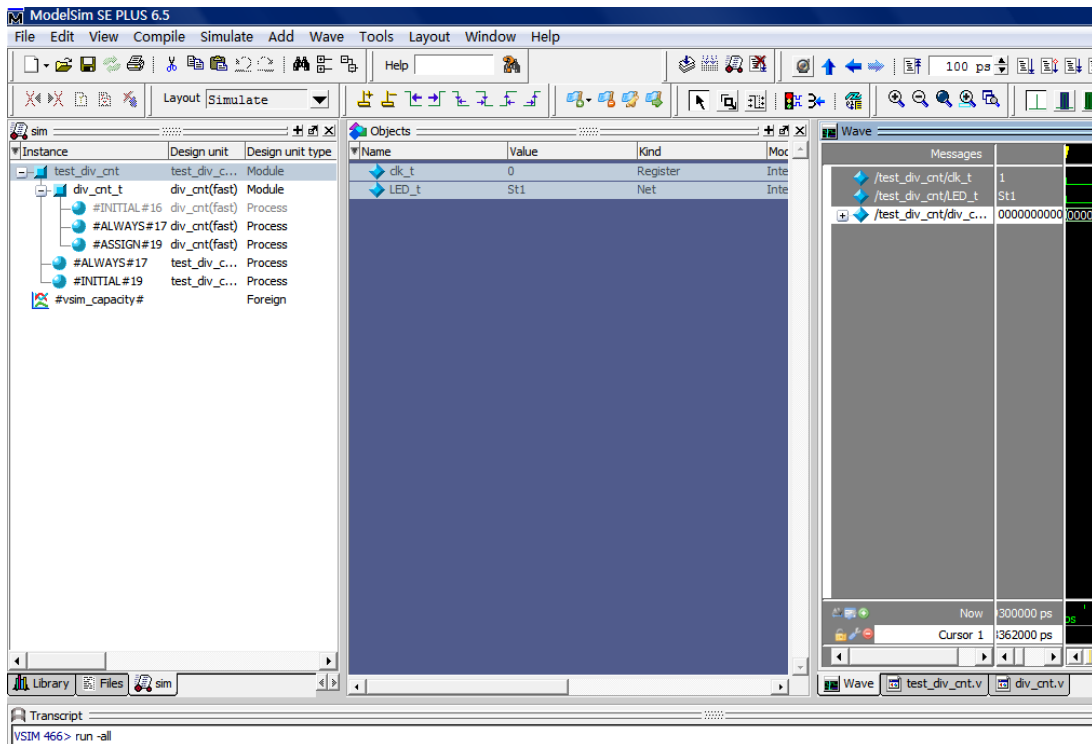
使用综合软件综合后生成的门级网表或者是实现后生成的门级模型进行仿真，不加入时延文件的仿真就是门级仿真。可以检验综合后或实现后的功能是否满足功能要求，其速度比代码功能仿真要慢，但是比时序仿真要快。

##### 时序仿真

在门级仿真的基础上加入时延信息的仿真就是时序仿真。优点是：比较真实的反映逻辑的时延与功能，缺点是速度比较慢，如果逻辑比较大，那么需要很长的时间。在这里仿真以 Altera 的器件为例。利用经过综合布局布线的网表和具有时延信息的反标文件进行仿真，可以比较精确的仿真逻辑的时序是否满足要求。

#### 2.modelsim菜单和工具栏介绍

典型的 modelsim 工作界面如图所示。在图的最上端为标题栏；下面一行为菜单栏；再下面为工具栏；左半部分为工作区（Workspace），在其中可以通过双击查看当前的工程及对库进行管理；右半部分为命令窗口区，在其中出现的命令行及提示信息称为脚本（Transcript）；最下面一行为状态栏。这里要注意的是，有些操作是无法通过菜单和工具栏来完成的，学习 ModelSim 一定要学会使用命令行方式来操作，常用的命令并不多，不是很难掌握，在后续章节将介绍仿真中的一些常见命令。因此，本节内容读者略读一下就可以了，实际试一下会更好。



## 标题栏

与一般的 Windows 窗口相同，界面的最上一行为标题栏，显示当前的应用程序的名称，通过点击标题栏的图标（或 Alt 键 + SpaceBar 空格键）可以对窗口进行诸如改变窗口大小、移动窗口位置、关闭窗口之类的操作，这些与 Windows 完全相同。

## 菜单栏

标题栏下方为菜单栏。菜单栏有八个菜单项，分别是：File（文件）、Edit（编辑）、View（视图）、Compile（编译）、Simulate（仿真）、Tools（工具）、Window（窗口）、Help（帮助）。下面分别罗列其具体选项。

### ● File（文件）菜单

文件菜单通常包含了对工程及文件等的操作。ModelSim 的文件菜单包含的命令有：New（新建）、Open（打开）、Close（关闭）、Import（导入）、Save（保存）、Delete（删除）、Change Directory（更改路径）、Transcript（对脚本进行管理）、Add to Project（为工程添加文件）、Recent Directories（最近几次的工作路径）、Recen Projects（最近几次工程）、Quit（退出）。

#### （1）新建文件命令（File/ New）

单击 File/ New 命令，将会出现一个子菜单，共包含四个选项：单击 Floder（新建文件夹）后，会出现对话框，提示输入新建的文件夹的名字，即可在当前目录下新建一个文件夹；单击 Source（新建源文件）后，会出现源文件类型的选项（VHDL，Verilog，Other），点击可分别新建对应格式的源文件；单击 Project（新建工程）后，会出现对话框，提示在 Project Name 处输入新建工程的名称，在 Project Location 处指定新建工程的存放路径，在 Default Library Name 处指明默认的设计库的名称，用户设计的文件将编译到该库中；单击 Library（新建一个库）后，会出现对话框，提示选择 Creat a New library and a logical mapping to it（新建一个库并建立一个逻辑映象）或 A map to an existing library（新建一个到已存在库的映象），在 Lirary name 处输入新建库的名称，在 Library phycial name 处输入存放库的文件名称。

#### （2）Open（打开文件）

单击会出现子菜单选择打开 File ( 文件 )、Project ( 工程 ) 及 Dataset ( WLF 文件 )。

( 3 ) Close ( 关闭 )

单击会出现子菜单选择关闭 Project ( 工程 ) 或 Dataset ( 仿真数据文件 )。

( 4 ) Import ( 导入 )

导入新的库, 在进行某些仿真时需要的一些库可以通过该方法导入, 根据提示指定源库路径及目标库路径, 一步步操作完成。注意 ModelSim 安装目录下的 modelsim.ini 文件不能为只读。在该文件中保存了 ModelSim 的一些设置信息, 后续章节将详细讨论该文件的内容及其含义。

( 5 ) Save ( 保存 )

保存当前仿真数据。

( 6 ) Delete ( 删除 )

删除指定的工程, 即删除.mpf 文件, mpf 是 ModelSim 工程的后缀名。

( 7 ) Change Directory ( 改变路径 )

改变当前工作路径, ModelSim 使用的是绝对路径, 而不是相对路径, 这与 ISE 不同, 在 ISE 中, 你可以将你的设计整个目录拷贝到其他任何地方, 只要目录完整, 你可以直接打开工程文件。而在 ModelSim 中, 若将整个目录拷贝到其他地方, 打开工程时其指向仍为原来工程的地址, 可以通过更改路径来设置新的路径。

( 8 ) Transcript ( 脚本 )

单击会出现子菜单选择操作 Save Transcript ( 保存主窗口中脚本 )、Save Transcript As ( 把主窗口中脚本另存为一个新文件 ) 或 Clear Transcript ( 清除主窗口中的脚本 )。

( 9 ) Add to Project ( 添加到工程 )

单击会出现子菜单选择操作 File ( 添加文件到当前工程 )、Simulation Configuration ( 添加设定的仿真配置 ) 或 Folder ( 添加文件夹 )。

( 10 ) Recent Directories ( 最近几次工作路径 )

可以从中选取最近几次的工作路径。

( 11 ) Recen Projects ( 最近几次工程 )

可以打开最近几次的工程。

( 12 ) Quit ( 退出 )

退出 ModelSim。

## ● Edit ( 编辑 ) 菜单

类似于 Windows 应用程序, 在编辑菜单中包含了对文本的一些常用的操作。

( 1 ) Copy ( 复制 )

复制选中的文档

( 2 ) Paste ( 粘贴 )

把剪切或复制的文档粘贴到当前插入点之前。

( 3 ) Select All ( 全选 )

选中主窗口中所有的抄本文档。

( 4 ) Unselect All ( 取消全选 )

取消已选文本的选中状态。

( 5 ) Find ( 查找 )

在命令窗口中查找字符或字符串。

## ● View ( 视图 ) 菜单

类似于其他 Windows 应用程序, 视图菜单可以控制在屏幕上显示哪些窗口。

#### (1) All Windows (所有窗口)

打开所有的 Model Sim 窗口, 你试一下该命令会发现 ModelSim 打开了许多窗口, 包括波形窗口、信号列表窗口、源文件窗口等等。

#### (2) Dataflow (数据流)

打开 Dataflow 窗口, 在该窗口中显示数据的流向。

#### (3) List (列表)

打开列表窗口。

#### (4) Process (进程)

打开过程窗口, 该窗口显示了设计中的进程所在的位置。

#### (5) Signals (信号)

打开信号窗口。该窗口显示了设计中所有信号的列表

#### (6) Source (源文件)

打开源文件窗口, 可以在源文件窗口中显示设计中使用的源文件。

#### (7) Structure (结构)

打开结构窗口, 该窗口以列表方式显示了设计中所有到的结构, 双击某一结构, 可以查找定义该结构的语句。

#### (8) Variables (变量)

打开变量窗口, 该窗口以列表方式显示了设计中定义的所有变量。

#### (9) Wave (波形)

打开波形窗口, 这是我们仿真时经常需要查看的窗口, 在其中显示了输入和输出的波形。

#### (10) Datasets

打开 Dataset 浏览器来打开、关闭、重命名或激活一个 Dataset。你在使用的时候会发现没有什么变化, 这时候你可以看看 Workspace 窗口下是不是多了一个选项卡。该选项卡显示的内容与 Structure 窗口显示的完全相同。

#### (11) Coverage (覆盖率)

查看仿真的代码覆盖率。

#### (12) Active Processes (活动的进程)

当前正在执行的进程。

#### (13) workspace (工作区)

打开当前的工作区。

#### (14) Encoding (编码)

以不同的编码查看。

#### (15) Properties

显示工作区中选中对象的属性。

### ● Compile (编译) 菜单

#### (1) Compile (编译)

把 HDL 源文件编译到当前工程的工作库中。

#### (2) Compile Options (编译选项)

设置 VHDL 和 Verilog 编译选项, 例如可以选择编译时采用的语法标准等。

#### (3) Compile All (全编译)

编译当前工程中的所有文件。

#### (4) Compile Select (编译选中的文件)

编译当前工程中的选中文件。

#### (5) Compile Order (编译顺序)



设置编译顺序, 一般系统会根据设计对 VHDL 自动生成编译顺序, 但对于 Verilog 需要指定编译顺序。

#### (6) Compile Report (编译报告)

有关工程中已选文件的编译报告。

#### (7) Compile Summary (编译摘要)

有关工程中所有文件的编译报告。

### ● Simulate (仿真) 菜单

这里的编译及运行命令类似于 VC 等高级语言的调试时候的命令。

#### (1) Simulate (仿真)

装载设计单元。

#### (2) Simulation Options (仿真选项)

设置仿真选项。

#### (3) Run (运行)

Run \*\*\*ns: 在该仿真时间长度内进行仿真。若要改变长度, 可在 Simulation Options 中设置或在工具栏中修改;

Run-All (运行所有仿真): 进行仿真, 直到用户停止它;

Continue (继续): 继续仿真;

Run-Next (运行到下一事件): 运行到下一个事件发生为止;

Step (单步): 单步仿真;

Step-Over: 仿真至子程序结束;

Restrat: 重新开始仿真, 重新加载设计模块, 并初始化仿真时间为零。

#### (4) Break (停止)

停止当前的仿真。

#### (5) End Simulation (结束仿真)

结束当前仿真。

### ● Tool (工具) 菜单

#### (1) Waveform Compare (波形比较)

在子菜单中有具体进行波形比较的命令。

#### (2) Coverage (覆盖率)

测试仿真的代码覆盖率, 所谓代码覆盖率是指仿真运行到当前已运行的代码占有所有代码的比例, 当然是越接近 100% 越好。

#### (3) Breakpoints (断点设置)

单击此选项出现断点设置对话框, 设置断点。

#### (4) Execute Macro (执行宏文件)

所谓的宏文件就是保存后的脚本, 脚本保存起来, 以后可以利用该命令来重新执行。

#### (5) Options (选项)

Transcript File: 设置脚本文件的保存。

Command History: 命令历史。

Save File: 保存脚本文件。

Saved Lines: 限制脚本文件的行数。

Line Prefix: 设置每一行的初始前缀。

Update Rate: 设置状态条的刷新频率。ModelSim Prompt: 改变 ModelSim 的命令提示符。

VSIM Prompt: 改变 VSIM 的命令提示符。

Paused Prompt: 改变 Paused 的命令提示符。

HTML Viewer: 设置打开在线帮助的文件。

(6) Edit Preferences (编辑参数选取) :

设置编辑参数。

(7) Save Preferences (保存参数选取) :

设置保存用的参数。

- Window (窗口) 菜单 :

(1) Initial Layout (初始化版面)

恢复所有窗口到初始时的大小和位置。

(2) Cascade (层叠)

使所有打开的窗口层叠。

(3) Tile Horizontally (水平平铺)

水平分隔屏幕, 显示所用打开的窗口。

(4) Tile Vertically (垂直平铺)

垂直分隔屏幕, 显示所用打开的窗口。

(5) Layout Style (版面格式)

Default (默认格式) : 与 Initial Layout 格式相同 ;

Classic (经典格式) : 采样低于 5.5 版本的格式 ;

Cascade : 与 Cascade 格式相同 ;

Horizontally : 与 Tile Horizontally 格式相同 ;

Vertically : 与 Tile Vertically 格式相同。

(6) Icon Children

除了主窗口之外的其他窗口缩为图标。

(7) Icon All

将所有窗口缩为图标。

(8) Deicon All

将所用缩为图标的窗口还原。

- Help (帮助) 菜单

(1) About ModelSim

显示 ModelSim 的版本、版权等信息。

(2) Release Notes

显示 ModelSim 的版本发布信息。

(3) Welocme Menu

显示欢迎画面。

(4) PDF Documentation

在子菜单中可以选择 ModelSim 的 PDF 文档。

SE HTML Documentation : ModelSim 的超文本文档。

(5) Tcl Help : Tcl 帮助文档。






















Tcl 是 Tools Command Language 的缩写, 它是一种可扩充的命令解释语言, 具有与 C 语言的接口和命令的能力, 应用非常广泛, 这方面也有专门的书籍。

(6) Tcl Man Pages : Tcl 主页面。

(7) Technotes : 技术文档。

## 工具栏

ModelSim 的工具栏提供了常用操作的快捷键, 如 打开、复制、粘贴、如何更新 ModelSim、编译选定、编译全部、仿真、停止仿真、重新开始仿真、仿真步长、运行一步、继续运行、运行所有、单步执行、主程序的单步执行。其最常用的几个功能按键如下所示。

- a) , , , , , , , ,  这几个都是非常常见的图标了, 分别是新建新的源文件, 打开文件对话框, 保存, 打印, 剪切, 复制, 贴粘, 撤消最后一步操作, 在当前窗口查找文本。在这里和下文提到的当前窗口, 均可靠点击某窗口的任何位置选择。当前窗口标题栏会以亮的蓝色显示。
- b) , 编译, 打开文件对话框, 选择 HDL 源文件, 把该源文件编译到当前工程的工作库中。
- c) , 全编译。编译当前工程中的所有文件。
- d) , 仿真。
- e) , 停止仿真。
- f) , 回到上一层。
- g) , 重新仿真。装载设计, 并将仿真复位到零, 重新仿真。
- h) , 设定单步仿真步长。
- i) , 运行当前仿真。在该仿真时间长度内进行仿真。
- j) , 继续仿真, 直到仿真结束, 或用户停止仿真。
- k) , 运行所有仿真, 直到仿真结束或用户停止仿真。
- l) , 添加一条坐标轴。
- m) , 删除一条坐标轴。

## 状态栏

ModelSim 的状态条如图所示, 其中 Project 后面为当前工程的名称, Now 后面为当前仿真时间。最右边的为选定的仿真结构中变量。

Project : NineBitAdder    Now: 100 ns    Delta: 0    sim:/add4in



## 实验结果：

利用 modelsim 对分频器的 Verilog 测试程序进行纯粹的功能仿真，综合后的功能仿真与布局布线后的时序仿真。

## 具体步骤：

### 1、先在 Quartus II 里生成一个例子程序

具体步骤略，可参考 Quartus 的具体步骤，例程代码见 div\_cnt.v。

```
module div_cnt(LED,clk);

input clk;
output LED;

reg [31:0] cnt = 0;
always @(posedge clk) cnt<=cnt+1;

assign LED = cnt[10];
endmodule
```

### 2 . 编写对应的testbench文件

```
module test_div_cnt;
reg clk_t;

wire LED_t;

always #50 clk_t = ~clk_t;

initial
begin
#0 clk_t = 0;
#100000000 $stop;
end

div_cnt div_cnt_t(LED(LED_t),clk(clk_t));
endmodule
```

### 3 . 纯功能仿真

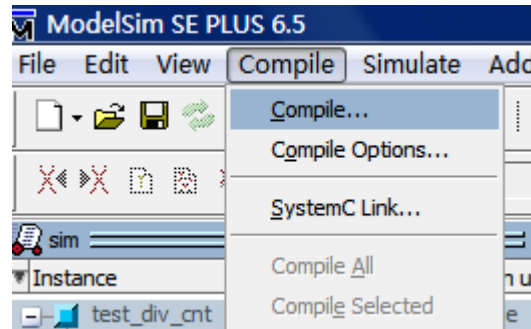
前面已经介绍过，具体的仿真可以分为三个阶段。一是纯粹的功能仿真；二是综合后的功能仿真；三是布局布线后的时序仿

真。下面就分别进行介绍。

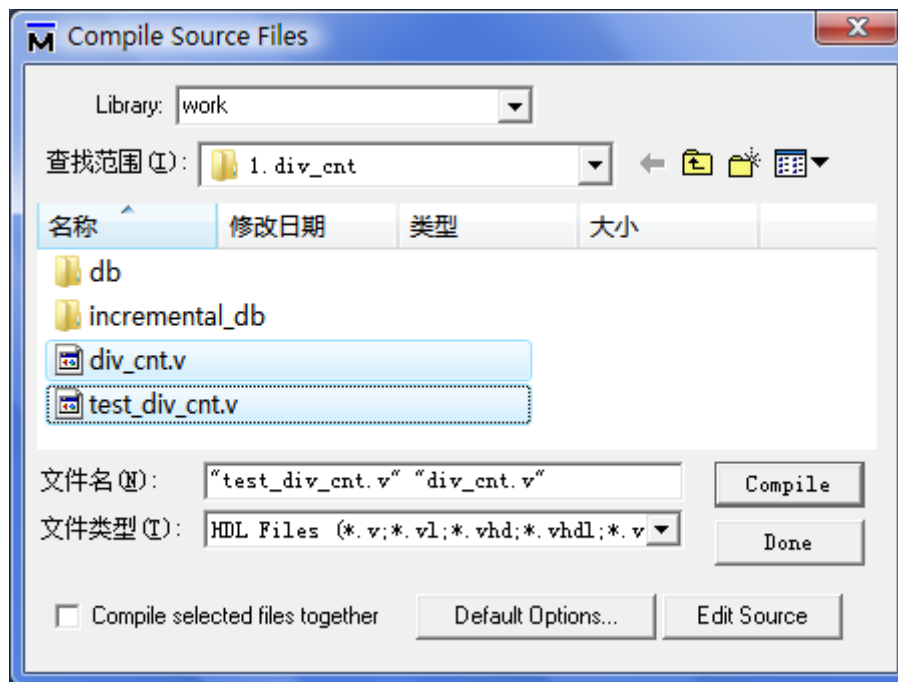
ModelSim 仿真有很多种流程，下面我们采用最简单的一种方式来进行仿真，其具体步骤如下

- 直接选择工程目录下的源文件与测试文件进行编译
- 启动仿真器，指定顶层设计单元
- 添加待观察信号
- 查看和调试结果

打开 modelsim，点击菜单栏上的 Compile 里的 Compile 选择项



选择刚才在 quartus 里建立的工程目录，将源程序与测试程序都选择上，点击 Compile

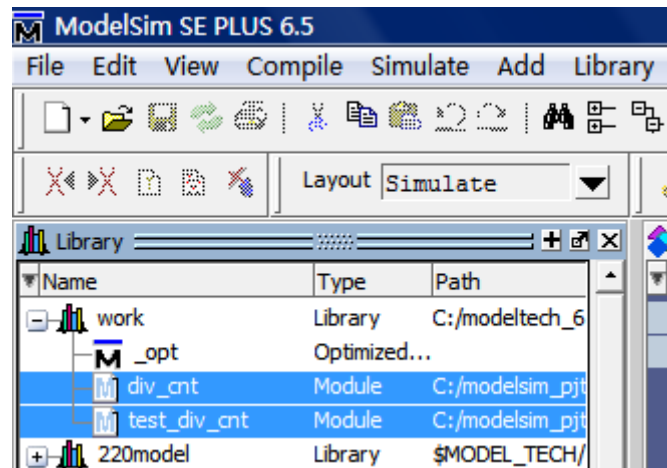


如果程序无误，则会在 Transcript 窗口显示如下信息：

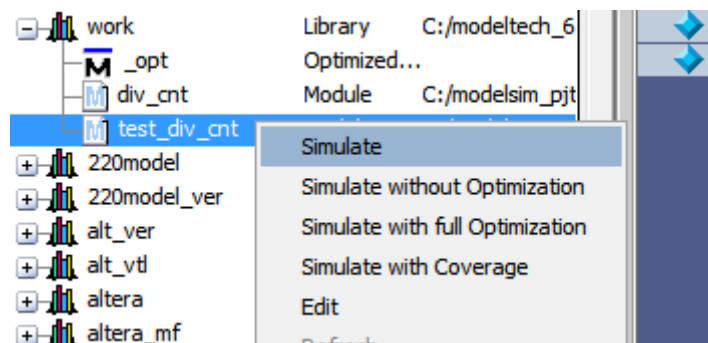
```
# -- Compiling module div_cnt
#
# Top level modules:
#   div_cnt
vlog -work work -L mtiAvm -L mtiOvm -L mtiUPF C:/modelsim_pjt/1.div_cnt/test_div_cnt.v
# Model Technology ModelSim SE vlog 6.5 Compiler 2009.01 Jan 22 2009
```

```
# -- Compiling module test_div_cnt
#
# Top level modules:
# test_div_cnt
```

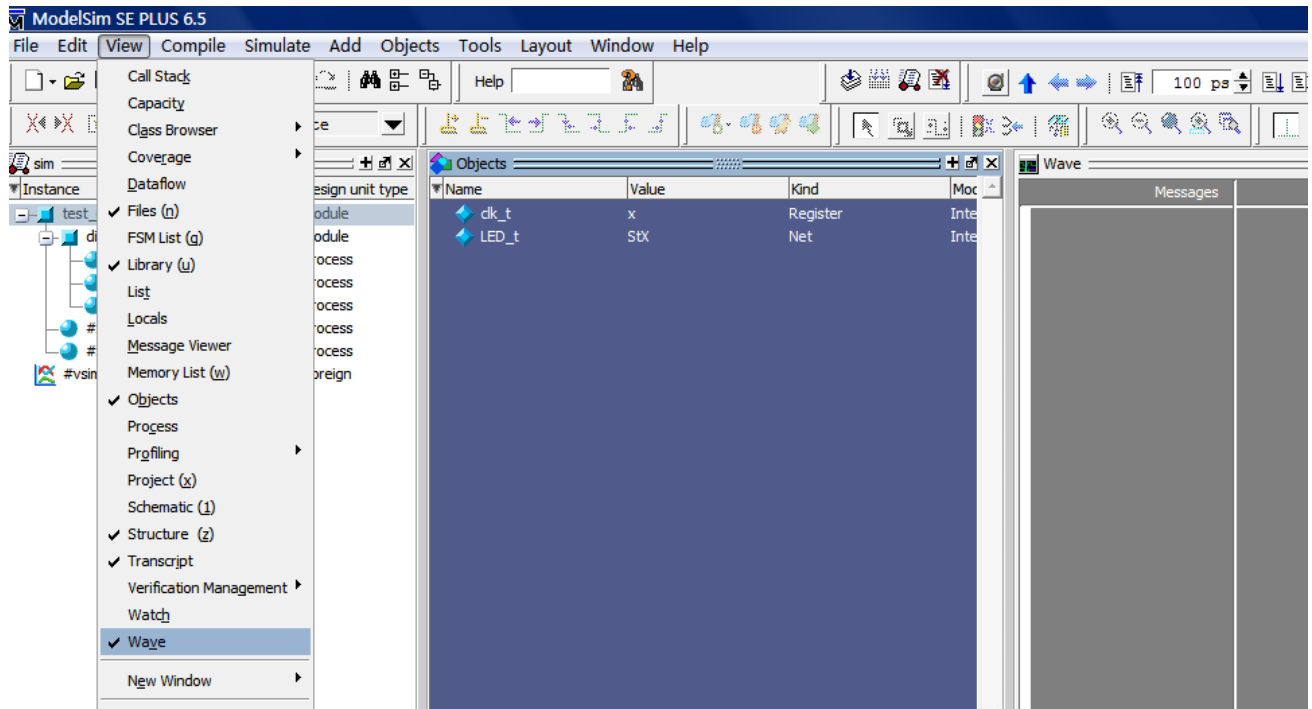
此时可以在 work 库中发现增加了如下两个模块，分别对应着程序的顶层模块与其对应的测试模块



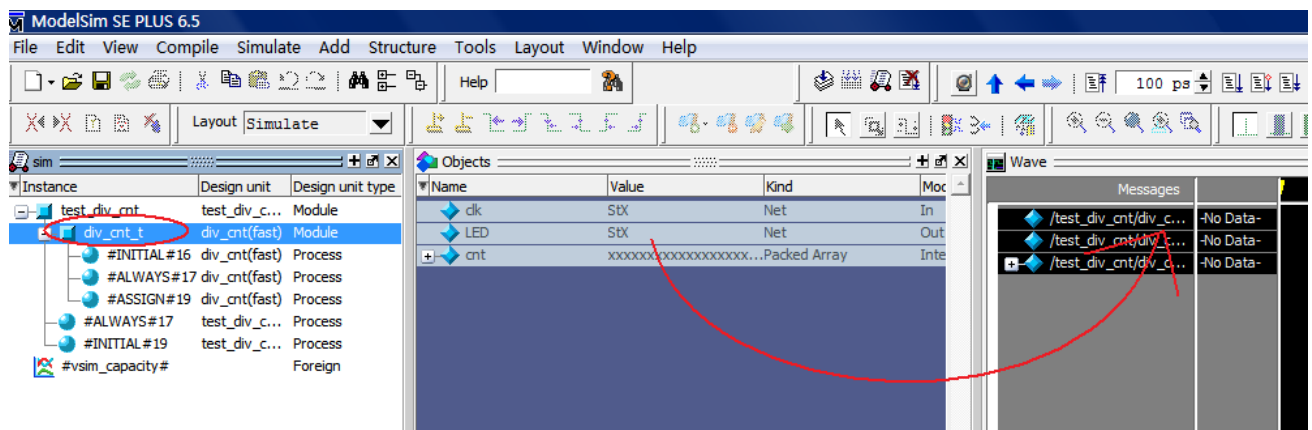
在测试模块名称上点击右键，选择 Simulate 进行仿真



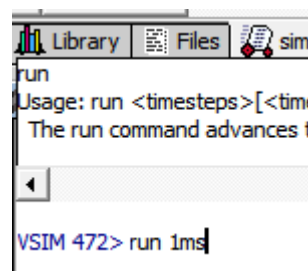
点击完毕后，确认一下 Objects 与 Wave 窗口是否已经打开，如果未打开，在 View 菜单下将其打开

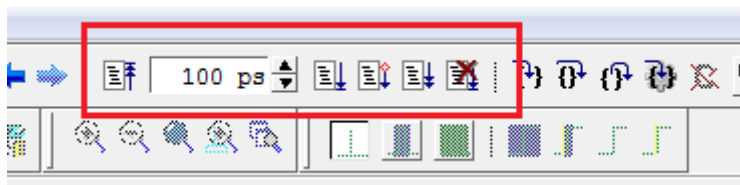



在 Instance 窗口中选择感兴趣的实体，并在 Objects 中选择需要观察的信号，拖拽到 Wave 窗口中，可以用 Ctrl 或 Shift 键多选

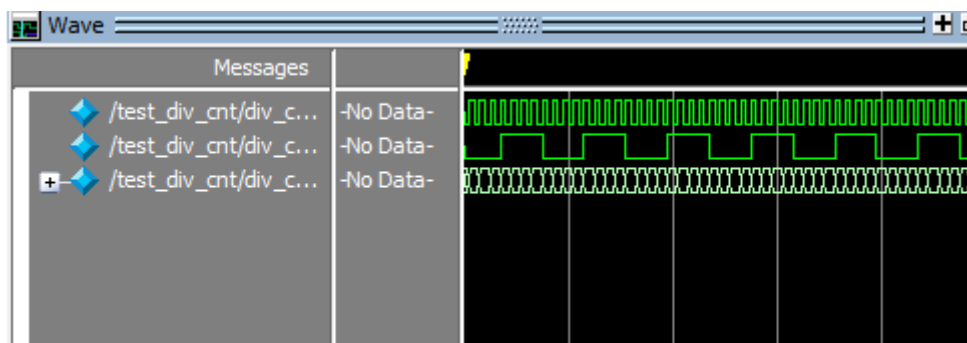



运行。在下面的命令行中输入运行的时间，回车。亦可以点击工具栏上的对应按钮执行仿真。

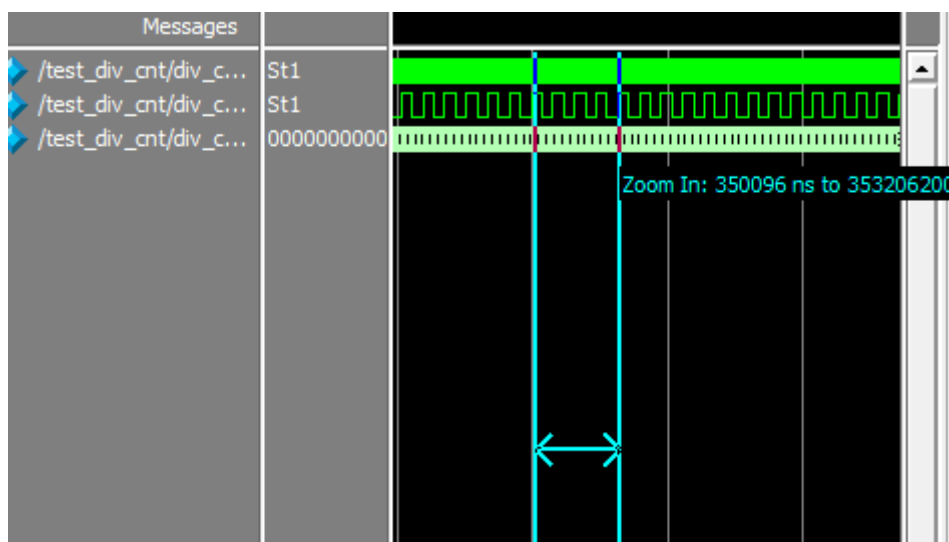




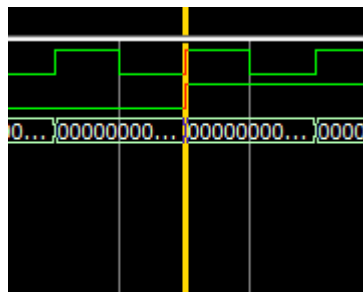
查看结果：点  图标，最大化波形窗口



点  图标，在波形窗口下，按住左键不放，向右下斜拉可以选择一个放大的区域，




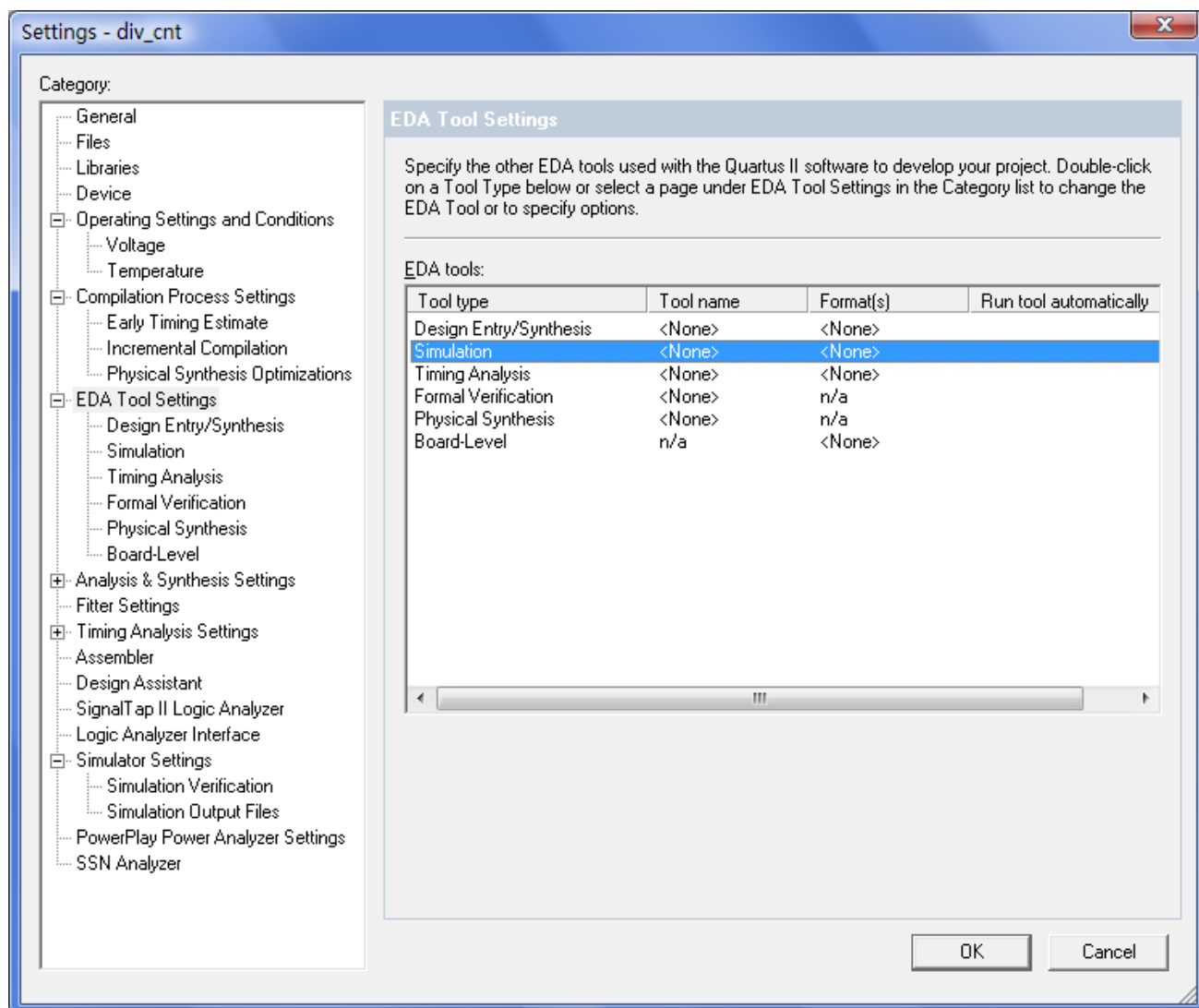
从仿真的结果来看，完全是没有延时的。这就是纯粹的功能仿真。



## 5. 综合后仿真

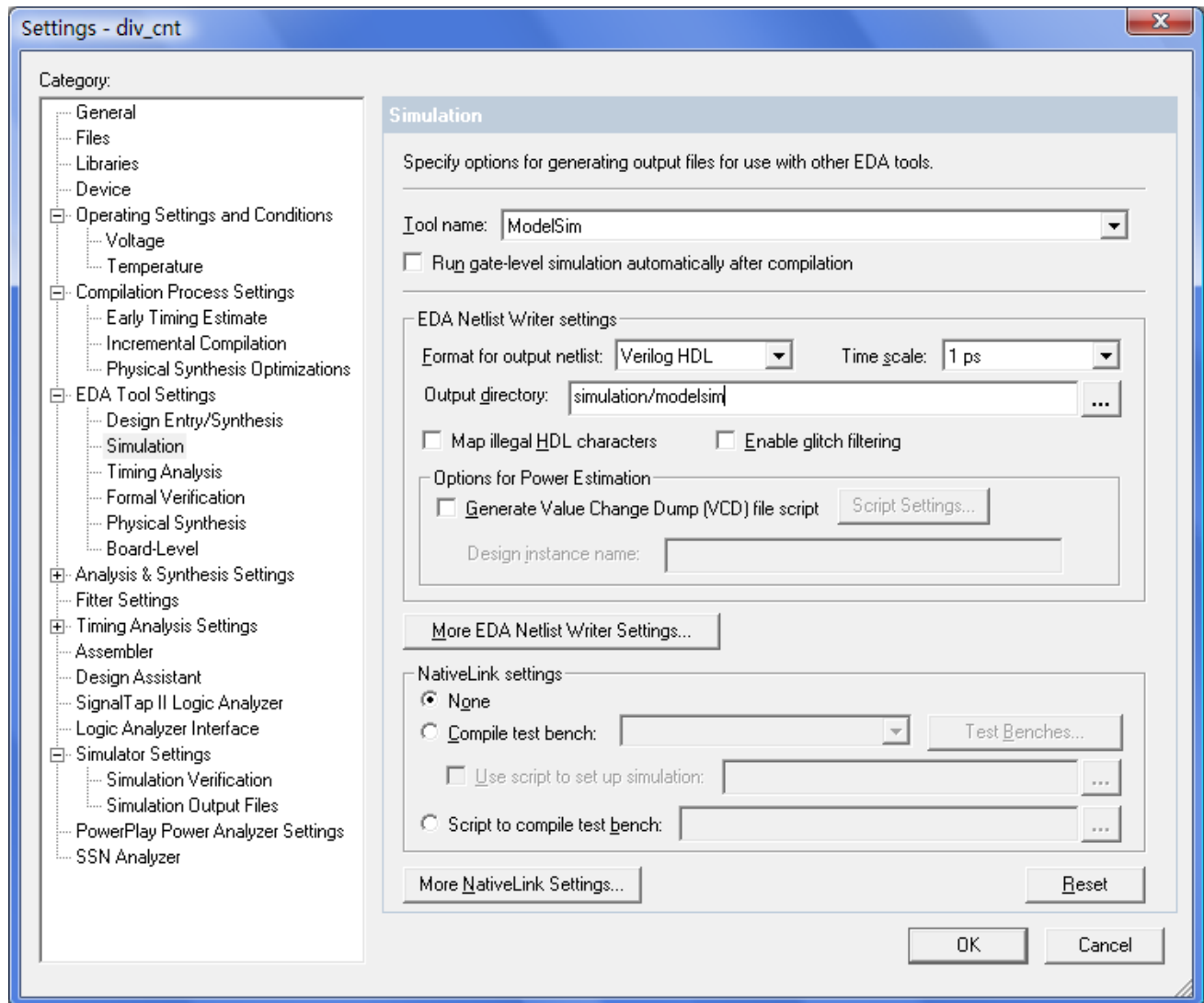
进行综合后功能仿真，我们当然需要综合器提供必要的的输入文件，即在 QuartusII 里面生成的网表文件；下面我们要回到 QuartusII，介绍一下在 QuartusII 里面生成网表文件的方法：

点 QuartusII 工具栏上  的图标，选 EDA Tool Setting, 双击 Simulation。

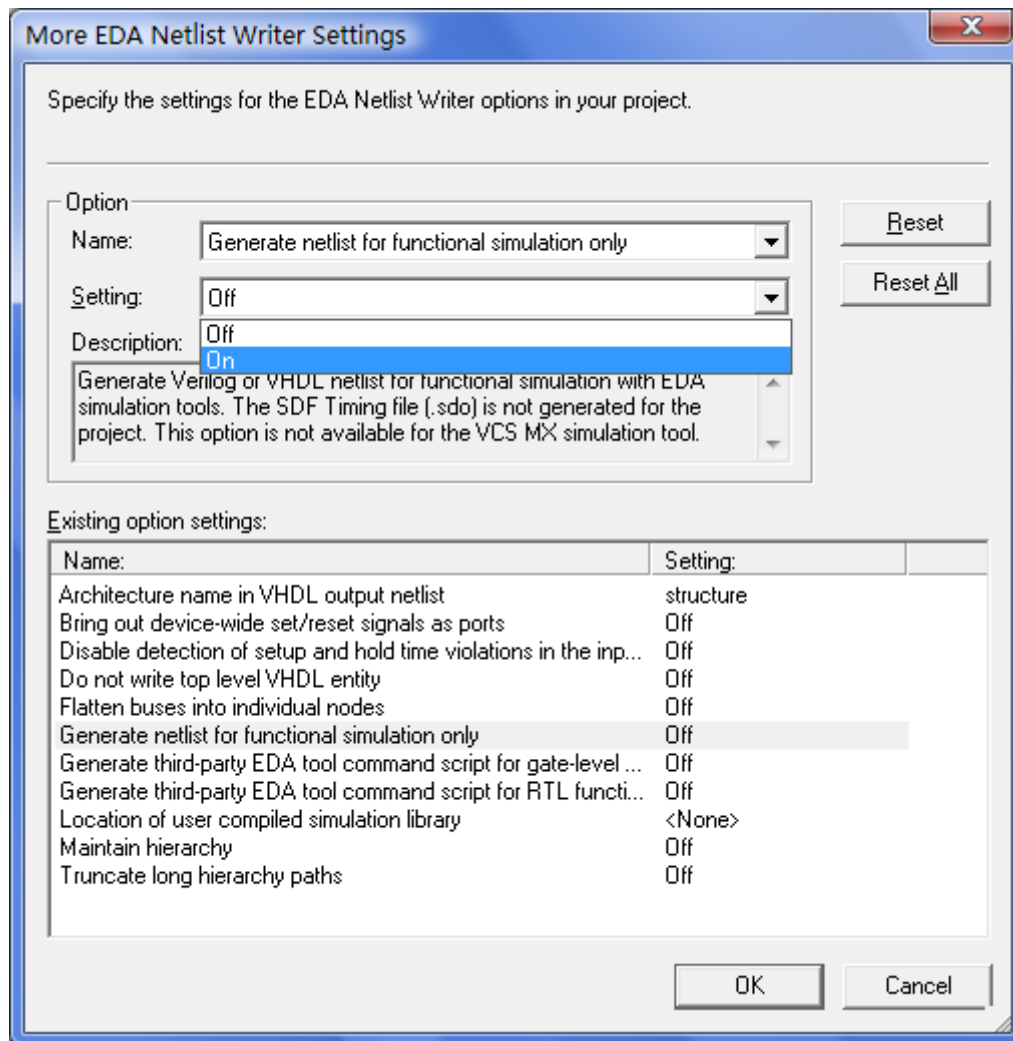


设置 ModelSim，输出网表的格式，以及网表文件保存的路径。

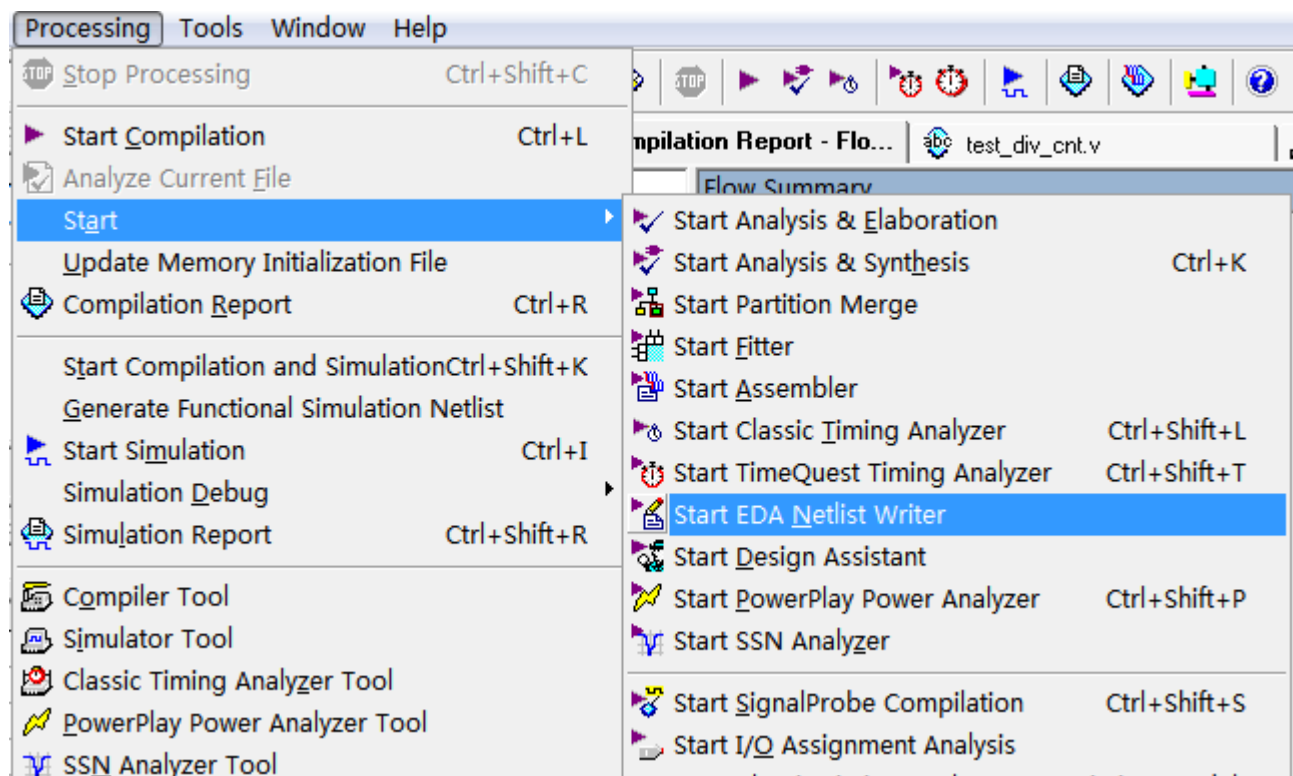




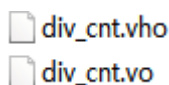
c) 点  , 将 Generate netlist for functional simulation only 设置为 ON。



点 OK，保存。点工具栏上的  图标，进行全编译。或者点击如下的选项，直接生成 EDA 网表，此时不需要全程编译

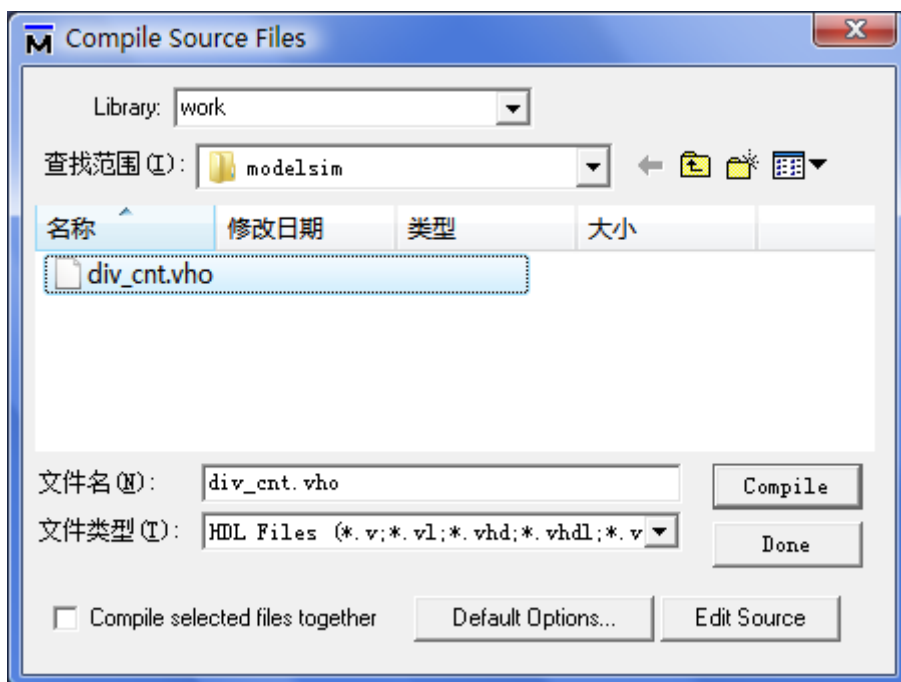


检查一下.../modelsim/ 里面，可以发现，已经生成了一个\*.vo 文件。

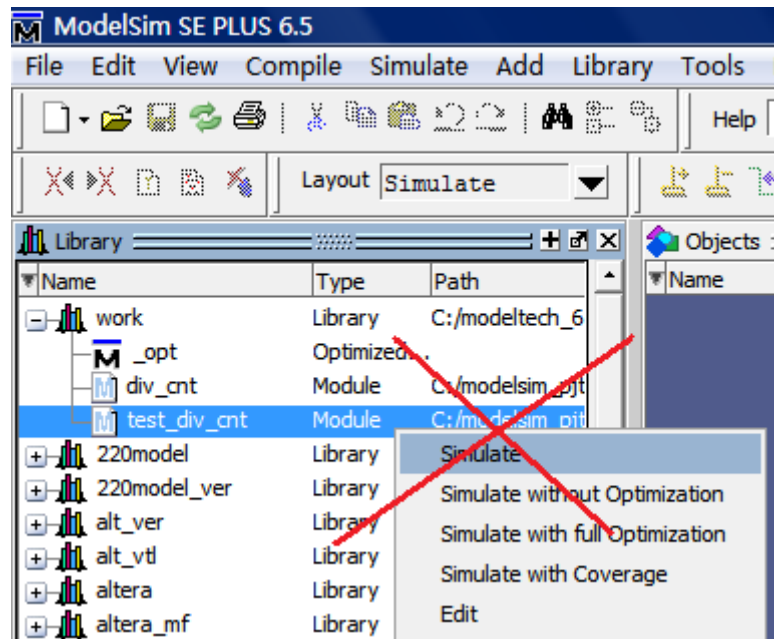


这个就是需要的网表文件。（如果是 VHDL 则生成的是\*.vho）

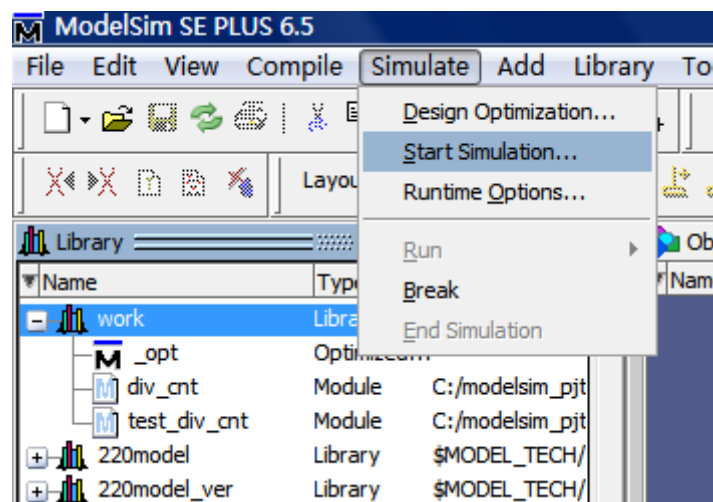
然后我们回到 modelsim，把刚才生成的 vho 或者 vo 文件编译进来



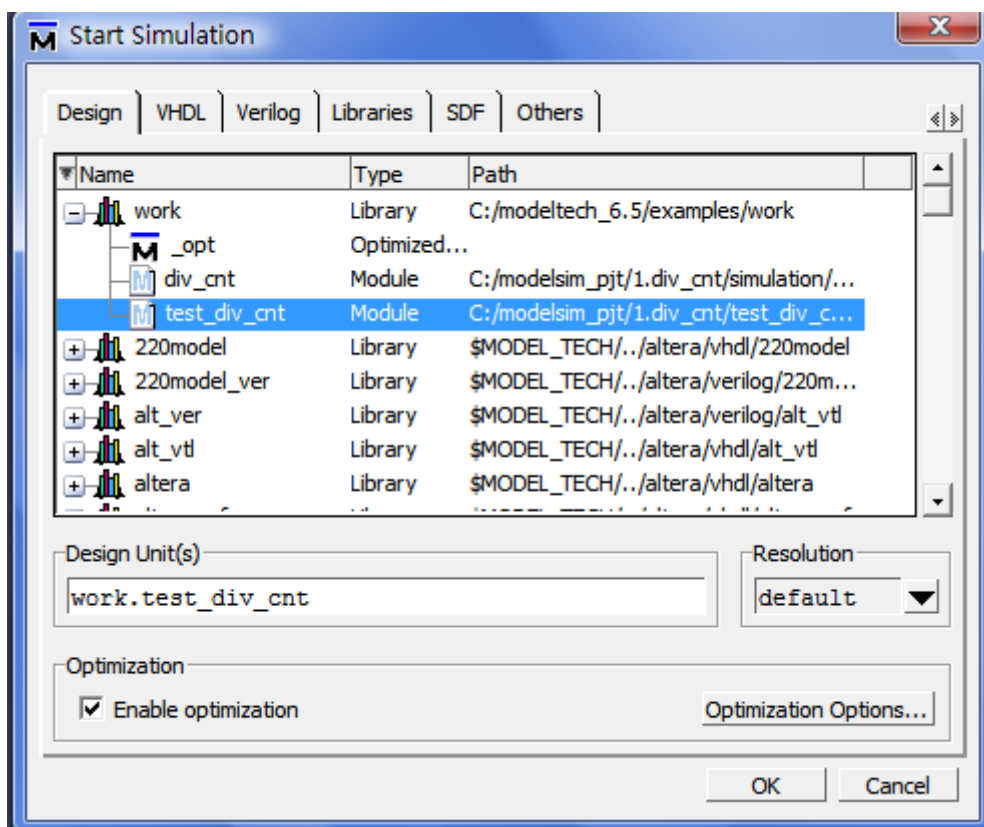
然后我们启动仿真器，这里注意，不要直接右键点击 Simulate



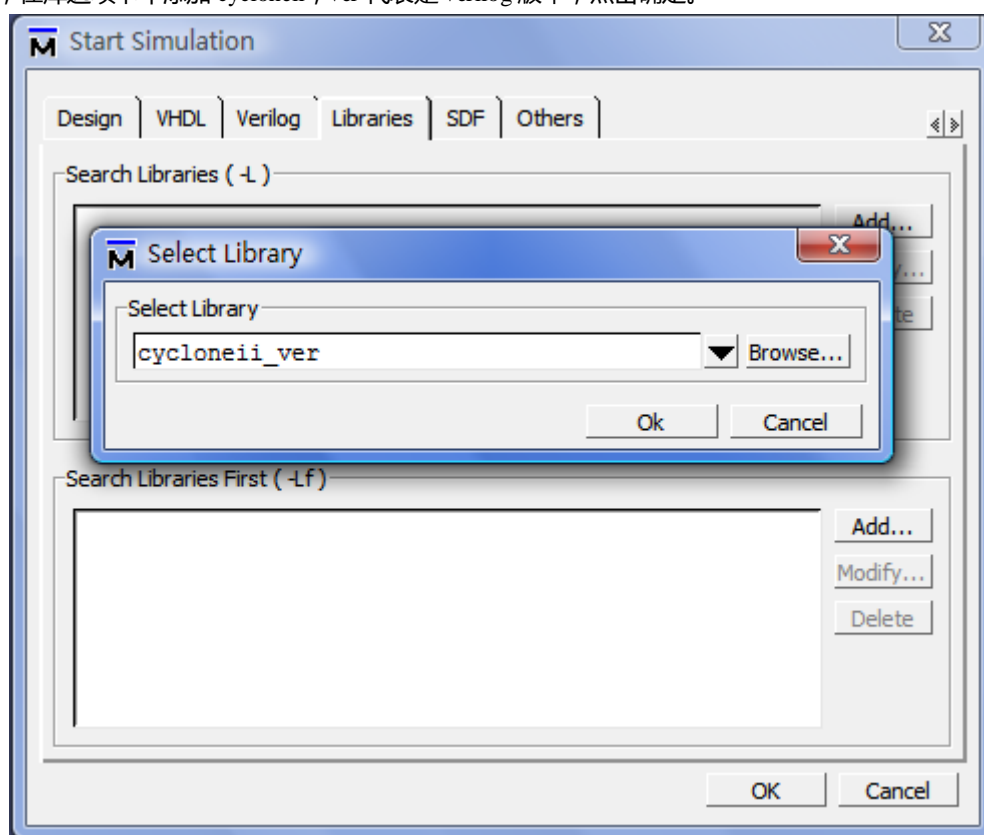
而在 Simulate 选项卡中选择 Start Simulate :



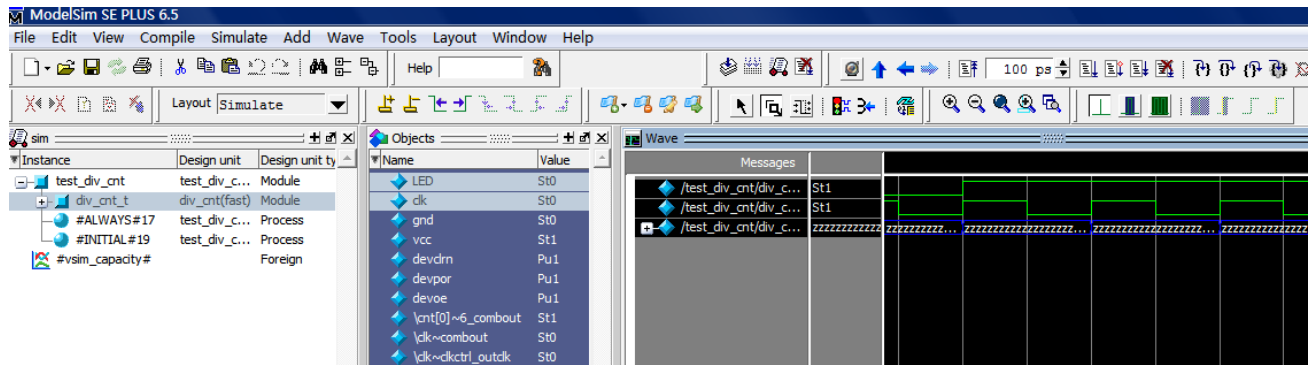
如下选择对应的测试实体



下面这一步注意，在库选项卡中添加 cycloneii，ver 代表是 verilog 版本，点击确定。



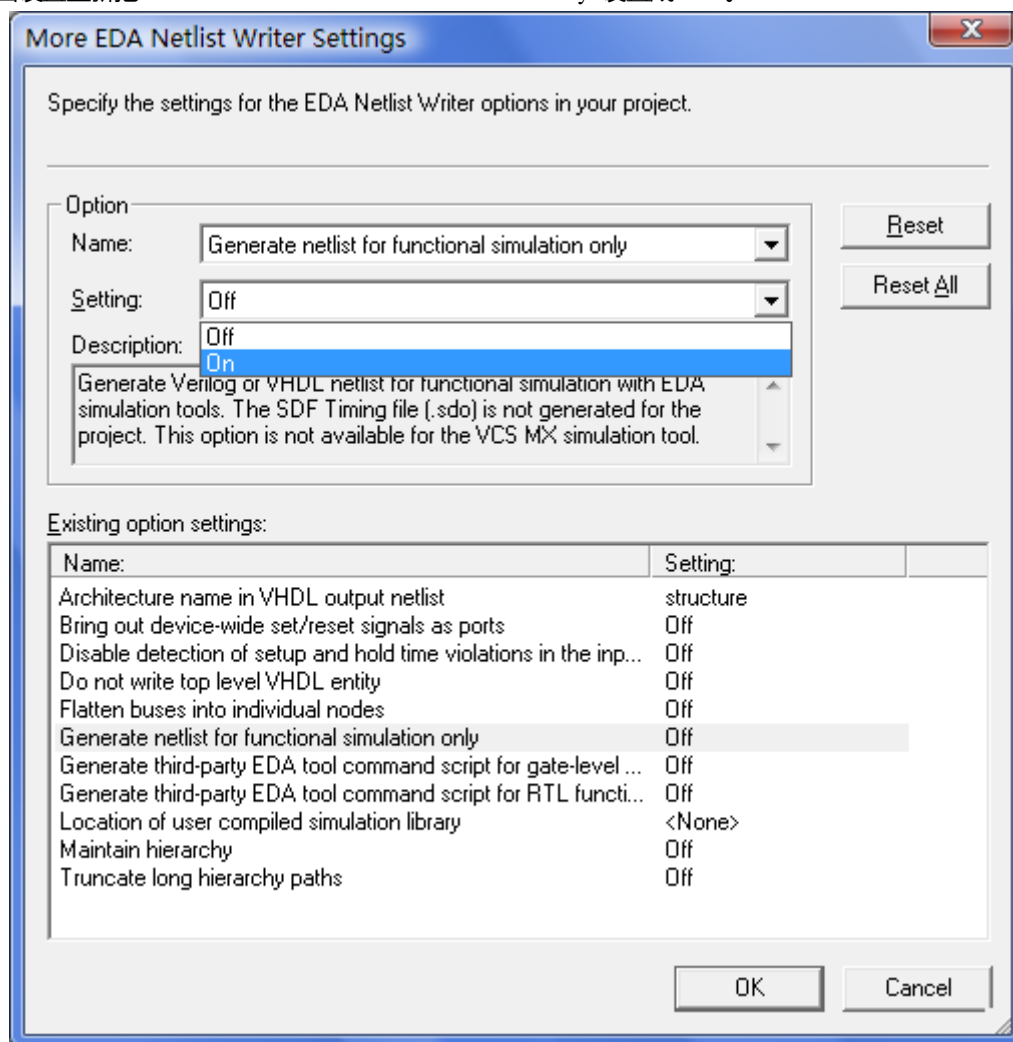
添加信号，执行仿真，方法就不重复了。可以发现，由于编译器提供了的 vho 附加了许多实际硬件电路的信息，Objects 中的选项也多了很多，这里我们仍然选择刚才一样的三个信号，最后查看和调试结果



可以看到, cnt 由于综合后被综合器优化, 导致其高位并没有完全使用, 所以显示为 z, 但输出时钟与输入时钟的边沿还是完全同步的。这说明综合后仿真只是考虑了生成硬件的原理结构, 但并没有考虑到布局布线的延时。

## 5. 布局布线后时序仿真

布局布线后时序仿真, 我们还需要给 modelsim 提供布局布线的延时信息文件, 这时候, 只需要将综合后仿真步骤中改变一步: QuartusII 里面设置重新把 Generate netlist for functional simulation only 设置成 OFF。

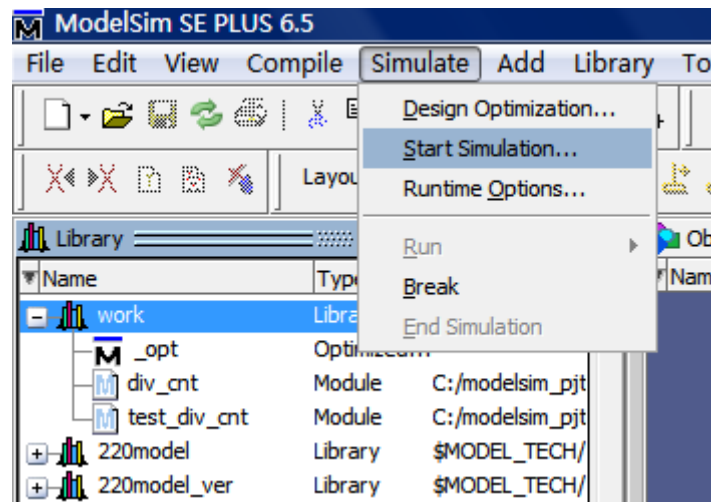


全编译后, 除了生成布局布线后网表文件以外, 还会生成延时文件\*.sdo。

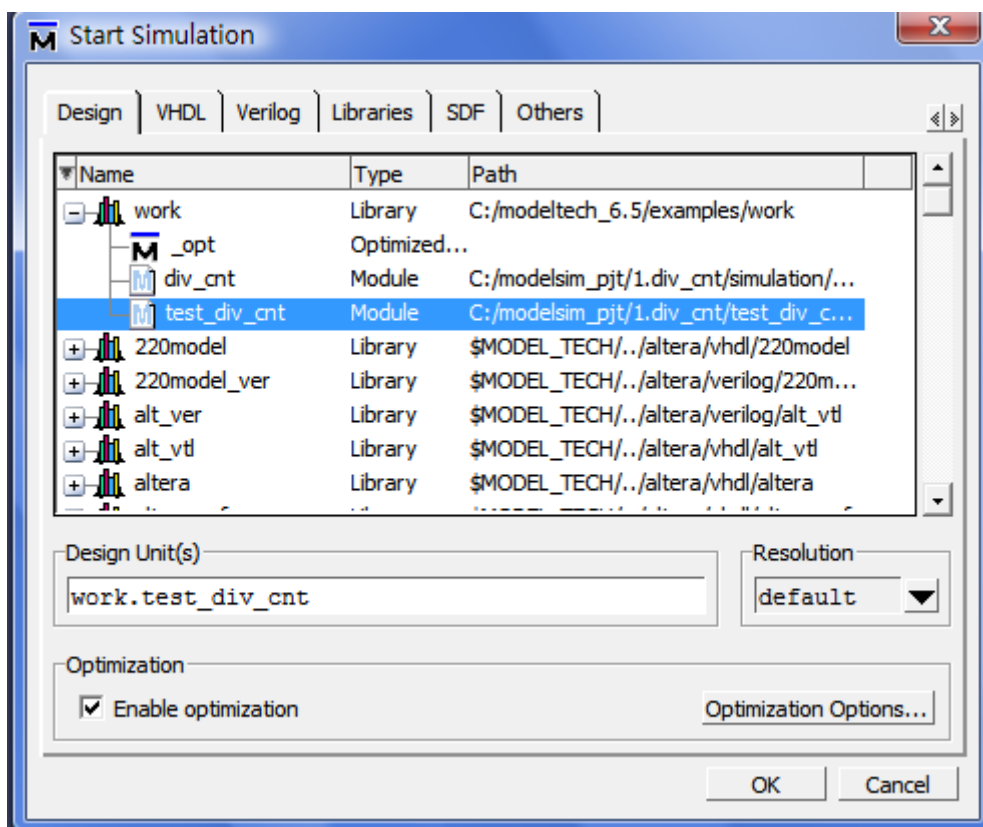


div\_cnt\_v.sdo

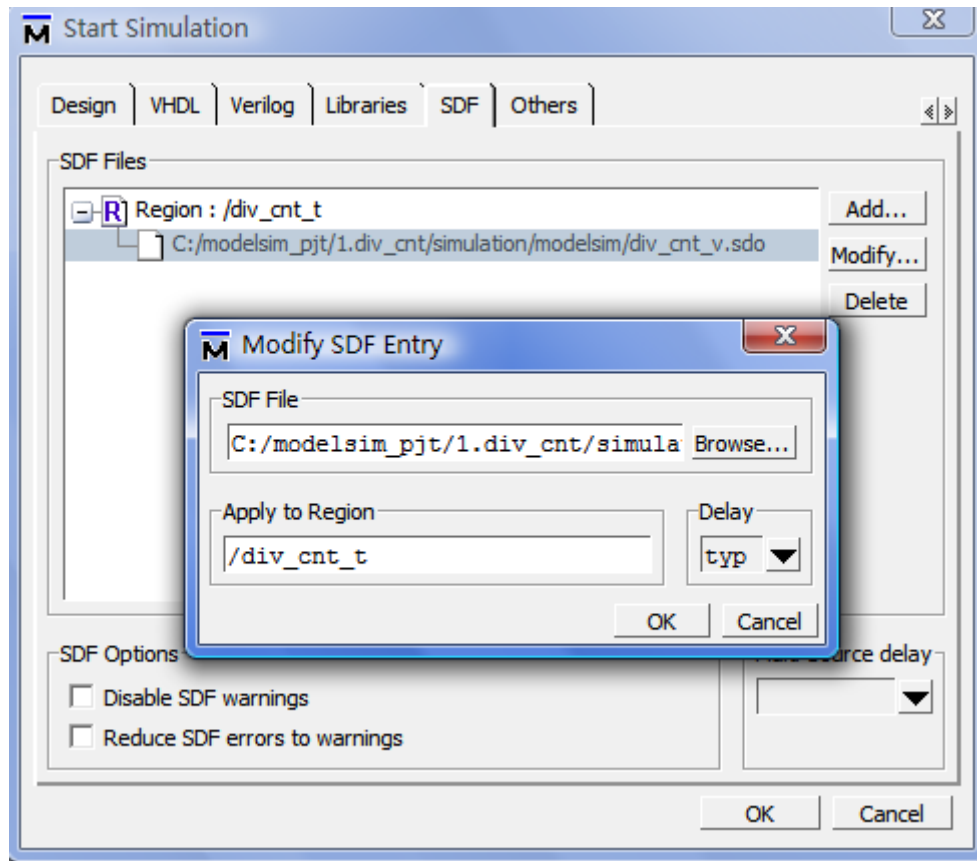
然后我们回到 modelsim，在 Simulate 选项卡中选择 Start Simulate：



如下选择对应的测试实体

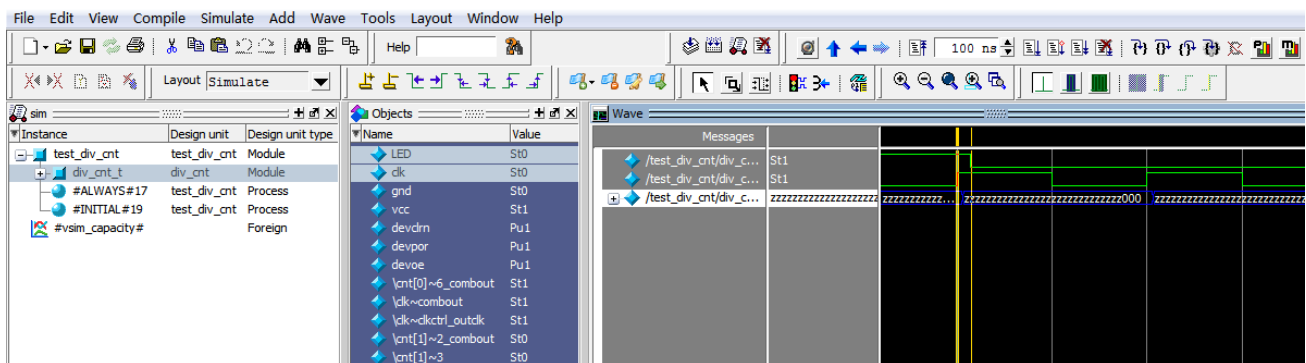


下面一步非常重要，加入 sdo 文件，注意指定 Apply to Region 里面，还要写 TestBench 里面例化顶层文件的例化名。



```
div_cnt div_cnt_t LED(LED_t),clk(clk_t));
```

添加信号，执行仿真，方法就不重复了。查看仿真波形，输出就显示了延时。



## 注意：

由于 modelsim 软件本身存在 bug，当按上面的设置无误后，出现无法解析 SDF 文件的时候，可以采用如下的方法解决。

```
# Loading cydoneii_ver.cydoneii_mux21
# Loading cydoneii_ver.cydoneii_dffe
# Loading cydoneii_ver.cydoneii_asynch_io
# Loading cydoneii_ver.cydoneii_dkctrl
# Loading cydoneii_ver.cydoneii_mux41
# Loading cydoneii_ver.cydoneii_ena_reg
# Loading instances from C:/modelsim_pjt/1.div_cnt/simulation/modelsim/div_cnt_v.sdo
# Loading cydoneii_ver.CYCLONEII_PRIM_DFFE
# Loading timing data from C:/modelsim_pjt/1.div_cnt/simulation/modelsim/div_cnt_v.sdo
# ** Warning: (vsim-SDF-3445) Failed to parse SDF file "div_cnt_v.sdo".
# Time: 0 ps Iteration: 0 Region: /test_div_cnt File: C:/modelsim_pjt/1.div_cnt/test_div_cnt.v
# Error loading design
```

新建一个 modelsim 的 Project。把除 sdf 文件以外的文件添加到工程目录下编译（注意需要将文件拷贝进来）。然后打开 start simulation 对话框，选 design 页，选中 testbench 文件，再点开 sdf 页，添加 sdf 文件（最好将 sdf 文件也拷贝到工程目录下），选中左下角的两个选项（禁止警告，把错误变成警告）。继续仿真即可。

## 实验总结：

呜呼~终于做完了今天的实验，真的不容易啊！步骤真的很繁多，而且 modelsim 做时序仿真的 BUG 也很让人头痛，不过一般情况下 Modelsim 只是拿来做功能仿真的，后仿真用的并不是很多。希望大家课后多多熟悉一下整个操作流程。

## 课后作业：

试着在 modelsim 中以新建工程的方式完成一遍今天的实验。（我们是通过直接编译进 work 库这种最简单的方式进行仿真的，其实也就是永远使用一个工作目录，而许多人仍然喜欢新建 modelsim 工程的方式来仿真，因为这样比较方便管理。但是直接编译的方式更加简单快捷，因为毕竟只是进行仿真验证，最终的工程仍然需要再 quartus 中实现的。）

## 相关信息

---

关于其他的相关信息，请访问以下网站

■ 购买本教程配套的开发套件，子卡或下载线缆：

<http://www.zr-tech.com>

■ 心得交流与问题互助：

<http://www.zr-tech.com/bbs>

## 版权信息

---

■ 本文档手册为ZRTech（[www.zr-tech.com](http://www.zr-tech.com)）原创资源，享有完全版权。

■ 任何收存和保管本文档各版本的单位和个人，未经本公司同意，不得随意复制、抄录、修改本文档的部分或者全部内容。

■ 转载本文档时请务必保证此文档的完整性。文档必须包含本版权信息。不得将转载作品以任何形式谋取商业利益，也不得向任何第三方提供，否则视为侵权。