

ModelSim 学习笔记(一)

初学 ModelSimSE 时 被迷糊了几天的若干概念

黄俊

April 2007

以前用的是 LATTICE 的，ispLEVER 有自带了一个 OEM 版的 ModelSim。要仿真时，不需要添加库，用起来比较方便，自己有点懒，所以就一直凑合着用。现在转向用 ALTERA 了，ALTERA 也有 OEM 版的 ModelSim，也不用添加库。后来听说 ModelSim SE 的功能更强大，速度更快，所以就决定把 ModelSim SE 好好摸索一下，再多学习一点关于 TestBench 技巧方面的知识。

我的学习资料主要是 ModelSim SE 自带的教程、ALTERA 提供的资料以及 edacn 上面 ModelSim 专栏由网友们上传的资料。因为是初学，加上看到英文资料一大堆，烦都烦死，而有些中文文档可能是有些步骤没有讲清楚，我实际按照文档上面说的一步一步做下来也老是完成不了，花了不少时间。我于是就想自己摸清楚后，把步骤截图下来，整理清楚，做成笔记。一方面加深自己的认识，另一方面对初学者也许会有些许用处。

我近期计划陆续整理出以下几个方面的学习笔记：

- 初学 ModelSimSE 时被迷糊了几天的若干概念
- 在 ModelSimSE 中添加 ALTERA 仿真库的详细步骤
- 用 ModelSimSE 进行功能仿真和时序仿真的方法（ALTERA 篇）
- ModelSimSE 中常用到的几个命令及 DO 文件的学习笔记
- 近来学到的几招 TestBench 的技巧

MSN: paulhuang_sz@hotmail.com

E-mail: huangjun5927@163.com

Blog: <http://www.edacn.net/index.php/2599>

初学 ModelSimSE 时被迷糊了几天的若干概念

- 1、什么是库？
- 2、什么是设计单元？
- 3、如何添加 ALTERA 仿真库？
- 4、ModelSim 的流程有哪些，实际开发中哪种流程最适用？
- 5、操作 ModelSim 要学习哪些操控方式？

1、什么是库(Library)?

(1) 我理解的库就是用来存放已编译好的设计单元 (Design Units) 的文件夹。

(2) 库的两种类型

- 工作库 (Work)
 - 存放当前设计文件编译后产生的设计单元
 - 编译前必须先创建好工作库
 - 每次编译只允许有一个工作库
 - 默认的工作库名是 work
- 资源库 (Resource)
 - 存放着所有可以被当前编译操作调用的已经编译过的设计单元
 - 每次编译允许同时调用多个资源库
 - ALTERA 的仿真库也属于资源库的一种
 - VHDL 的库可以直接通过 LIBRARY 和 USE 语句直接调用, 例如:
在 QuartusII 里面可以产生用于仿真的网表文件. vho 文件中就会带下面这样的语句:

```
LIBRARY IEEE, cycloneii;
USE IEEE.std_logic_1164.all;
USE cycloneii.cycloneii_components.all;
```

(3) 工作库和资源库其实用图 1 就可以直接分辨出来:

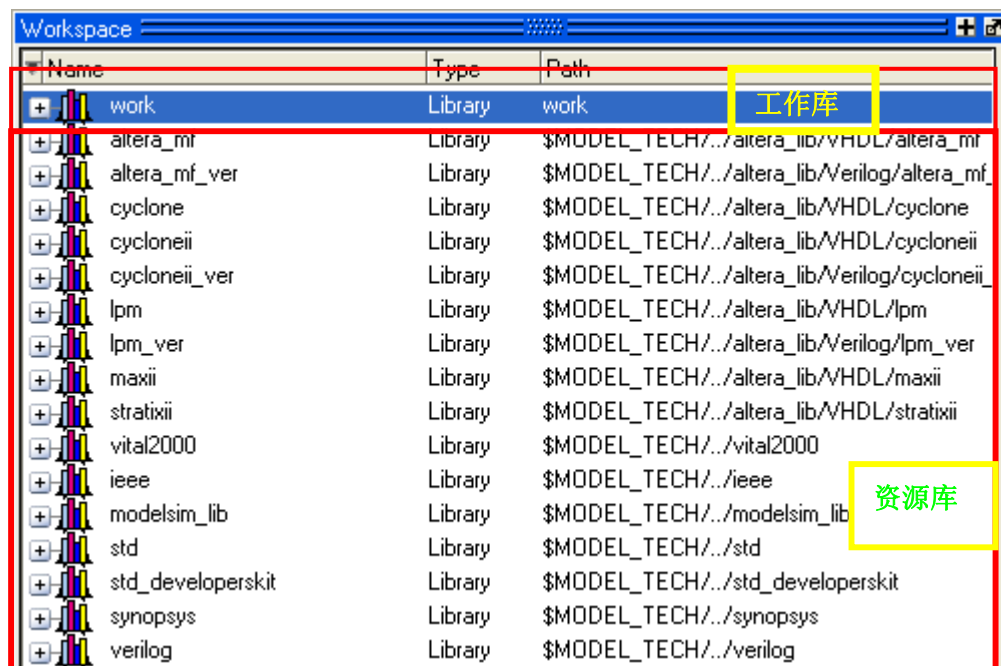


图 1 工作库和资源库

2、什么是设计单元 (Design Units) ?

(1) HDL 文件经过编译后, 会以设计单元的形式存放在库中, 设计单元可以是以下模块:

■ VHDL

- ◆ Entity
- ◆ Package Declaration
- ◆ Configuration
- ◆ Architecture (次级设计单元, 一个库中可以有多个同名的次级设计单元, 仅 VHDL 有次级设计单元)
- ◆ Package body (次级设计单元, 一个库中可以有多个同名的次级设计单元, 仅 VHDL 有次级设计单元)

例如:



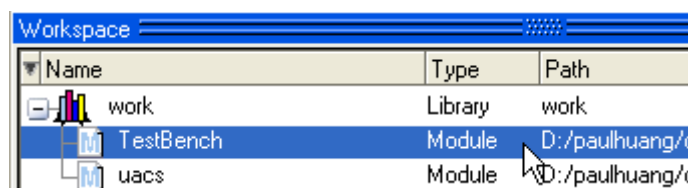
Name	Type	Path
altera_mf	Library	\$MODEL_Tt
a_graycounter	Entity	E:/altera/61...
alt3pram	Entity	E:/altera/61...
altaccumulate	Entity	E:/altera/61...
altcam	Entity	E:/altera/61...

图 2 Entity 设计单元实例

■ Verilog

- ◆ Module
- ◆ User Defined Primitive

例如:



Name	Type	Path
work	Library	work
TestBench	Module	D:/paulhuang/c...
uacs	Module	D:/paulhuang/c...

图 3 Module 设计单元实例

3、如何添加 ALTERA 的仿真库?

搞清楚这个问题花了不少时间, 详细的步骤记录在另一篇笔记《在 ModelSimSE 中添加 ALTERA 仿真库的详细步骤》中。

4、ModelSim 的流程有哪些，实际开发中哪种流程最适用？

我看了 ModelSim 自带的教程，主要有两种流程。

(1) 最基本的流程

- Step1 建库(library) (一个或多个)
- Step2 映射库到目录(directory)
- Step3 编译设计文件
- Step4 启动仿真器，指定顶层设计单元
- Step5 查看和调试结果

(2) 基于工程(Project)的流程

- Step1 新建一个工程
- Step2 添加文件到工程中去
- Step3 编译设计文件
- Step4 启动仿真器，指定顶层设计单元
- Step5 查看和调试结果

我觉得实际开发中用第二种流程最好。因为基于工程好处多一些：创建一个工程会自动建一个 Work 库，如果资源库（主要是仿真库）已经建好了的话，就不需要理会太多库的问题；工程文件(.mpf)包含了所有的在 ModelSim 的初始化文件 modelsim.ini 中的设置。可以很方便地对设计文件进行管理，可以方便地对编译进行设置，可以非常容易共享库文件而不需要拷贝文件到一个本地的目录下。

详细的仿真流程记录在另一篇笔记《用 ModelSimSE 进行功能仿真和时序仿真的方法 (ALTERA 篇)》当中。

5、操作 ModelSim 要学习哪些操控方式？

(1) 图形化用户接口 (GUI)

- i. 在这个界面下也可以接受命令行输入。
- ii. 重点推荐使用这种方式

(2) 互动型命令行形式 (CMD)

- i. 接口仅是一个命令控制行
- ii. 没有用户接口

(3) TCL 脚本以及 ModelSim 宏

- i. TCL - 工业标准的脚本语言宏（读音为“踢扣”）
- ii. (DO 文件) - 使用起来很方便，建议掌握

(4) 批处理

- i. 我暂时不学

ModelSim 学习笔记(二)

在 ModelSimSE 中
添加 ALTERA 仿真库的
详细步骤

黄俊

April 2007

以前用的是 LATTICE 的，ispLEVER 有自带了一个 OEM 版的 ModelSim。要仿真时，不需要添加库，用起来比较方便，自己有点懒，所以就一直凑合着用。现在转向用 ALTERA 了，ALTERA 也有 OEM 版的 ModelSim，也不用添加库。后来听说 ModelSim SE 的功能更强大，速度更快，所以就决定把 ModelSim SE 好好摸索一下，再多学习一点关于 TestBench 技巧方面的知识。

我的学习资料主要是 ModelSim SE 自带的教程、ALTERA 提供的资料以及 edacn 上面 ModelSim 专栏由网友们上传的资料。因为是初学，加上看到英文资料一大堆，烦都烦死，而有些中文文档可能是有些步骤没有讲清楚，我实际按照文档上面说的一步一步做下来也老是完成不了，花了不少时间。我于是就想自己摸清楚后，把步骤截图下来，整理清楚，做成笔记。一方面加深自己的认识，另一方面对初学者也许会有些许用处。

我近期计划陆续整理出以下几个方面的学习笔记：

- 初学 ModelSimSE 时被迷糊了几天的若干概念
- 在 ModelSimSE 中添加 ALTERA 仿真库的详细步骤
- 用 ModelSimSE 进行功能仿真和时序仿真的方法（ALTERA 篇）
- ModelSimSE 中常用到的几个命令及 DO 文件的学习笔记
- 近来学到的几招 TestBench 的技巧

MSN: paulhuang_sz@hotmail.com

E-mail: huangjun5927@163.com

Blog: <http://www.edacn.net/index.php/2599>

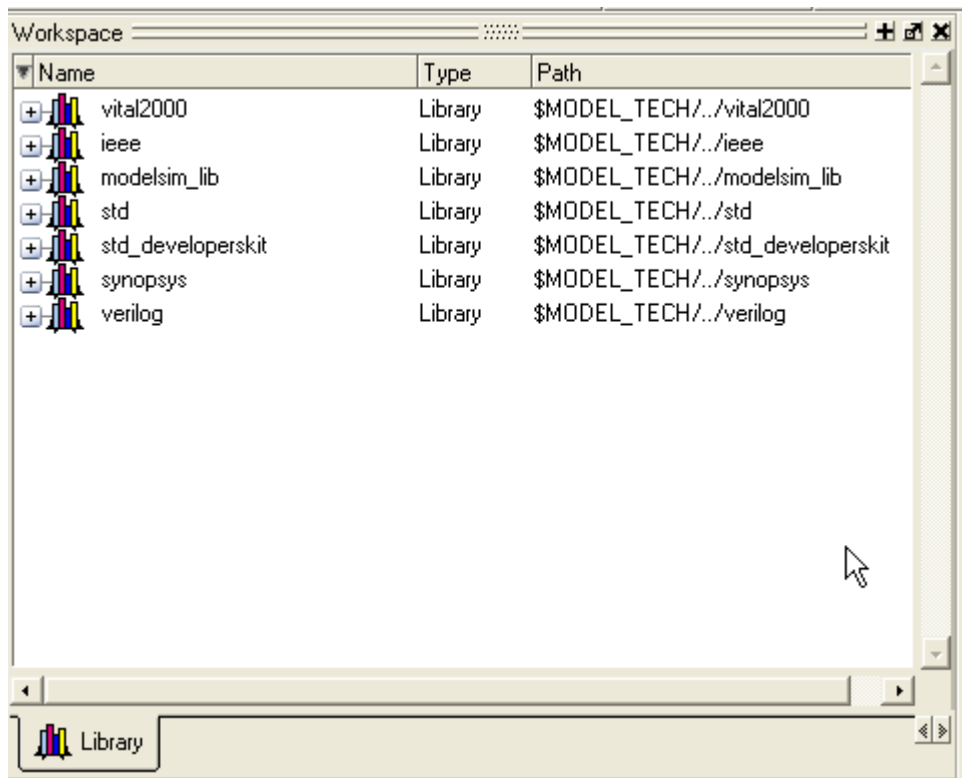
在 ModelSimSE 中添加 ALTERA 仿真库的详细步骤

装仿真库前要先了解几个概念

- 1、装 ModelSim 之前，要先装 QuartusII。安装好 QuartusII 后，在其安装目录下...\quartus\eda\sim_lib 里面存放了所有的仿真原型文件（simulation model files）。每个 PLD 厂家的开发软件装好后都有相应的目录存放这些仿真原型文件。VHDL 和 Verilog 的仿真原型文件不一样，VHDL 的会麻烦一些，一般都会多一个文件，并在编译时有仿真顺序的要求！
- 2、一个工程里面，资源库可以同时有多个，PLD 厂家的仿真库其实可以看成资源库的一种。我们要建四种库：（1）lpm，调用了 lpm 元件的设计仿真时需要；（2）altera_mf，调用了 Altera 的 MegaFunction 的设计的仿真时需要；（3）altera_primitive，调用了 Altera 的原语（primitive）的设计的仿真时需要；（4）元件库，例如 cyclone。在仿真中必用的特定型号的 FPGA/CPLD 的库。前三种库是调用了相应的 ALTERA 设计模块的设计进行仿真时必备的库。第四种库是进行综合后功能仿真和布线后时序仿真必备的库。关于库的笔记在《初学 ModelSimSE 时被迷糊了几天的若干概念》中有记录。
- 3、用户新建资源库和工作库是基于工程（project）的，新建工程后了，原来新建的库就会统统不见了。因此，就有必要修改 ModelSim 一个很重要的初始化文件——modelsim.ini。该文件位于 ModelSim 安装目录下。

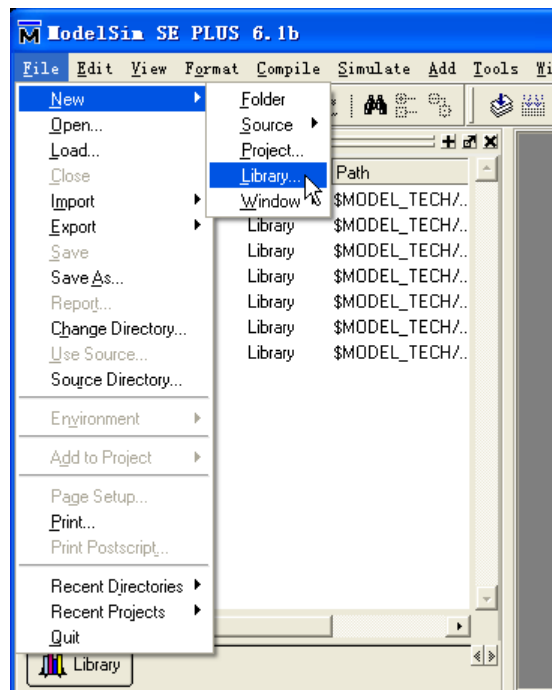
开始吧，步骤如下：

- 1、新建一个文件夹，这个文件夹是用来存放 ALTERA 编译后的库文件的。
注意：路径不要有中文名和空格。我就直接在 ModelSim 的安装目录下新建这个文件夹，命名为 altera_lib。因为仿真 VHDL 和 Verilog 的库是不同的，如果两种语言的库都要建的话，就再分细一点，在 altera_lib 里面再建 VHDL 和 Verilog 两个文件夹。
- 2、打开 ModelSim SE，刚刚装好的 ModelSim SE 默认的工作目录(Directory)是 ModelSimSE 安装目录下的 examples。此时，可以看到库里面最原始的库如下图所示：

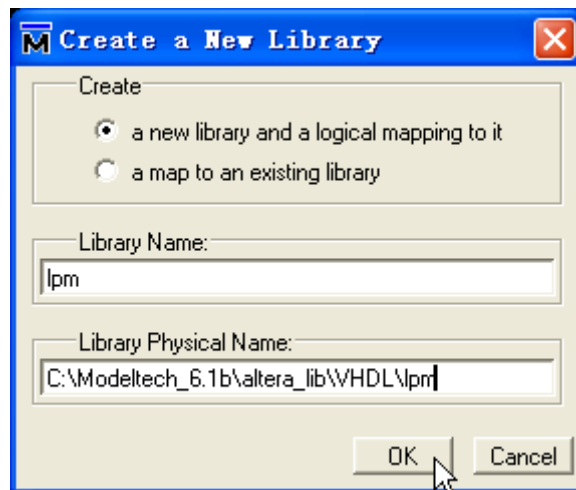


下面以新建 VHDL 的 **lpm** 库为例

3、先新建一个 Library, **File→New→Library...**



4、在 **Library Name** 里面输入库名 **lpm**, 在 **Library Physical Name** 里面输入该库存放的路径。例如: C:\Modeltech_6.1b\altera_lib\VHDL\lpm:

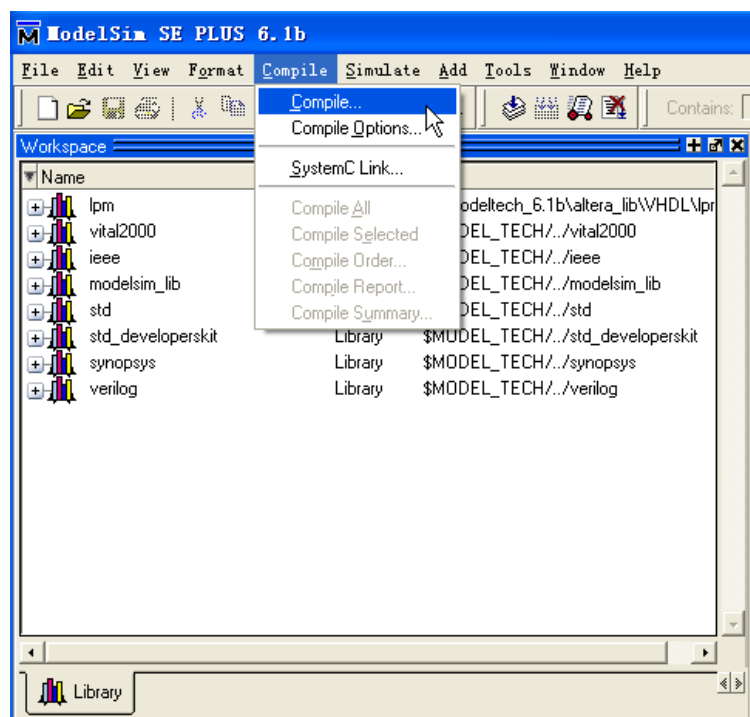


注：如果要进行综合后仿真和布线后的时序仿真，VHDL 的库名最好要取 altera_mf, lpm, cyclon, cycloneii, Verilog 的库名最好取 altera_mf_ver, lpm_ver, cyclone_ver, cycloneii_ver 这样的标准名字。特别是基于 VHDL 的设计更是这样，因为 VHDL 的库可以直接通过 LIBRARY 和 USE 语句直接调用，而 QuartusII 里面生成的用于仿真的输出网表文件都与这些库的名字有关例如在 QuartusII 里面可以产生用于仿真的网表文件. vho 文件中就会带下面这样的语句：

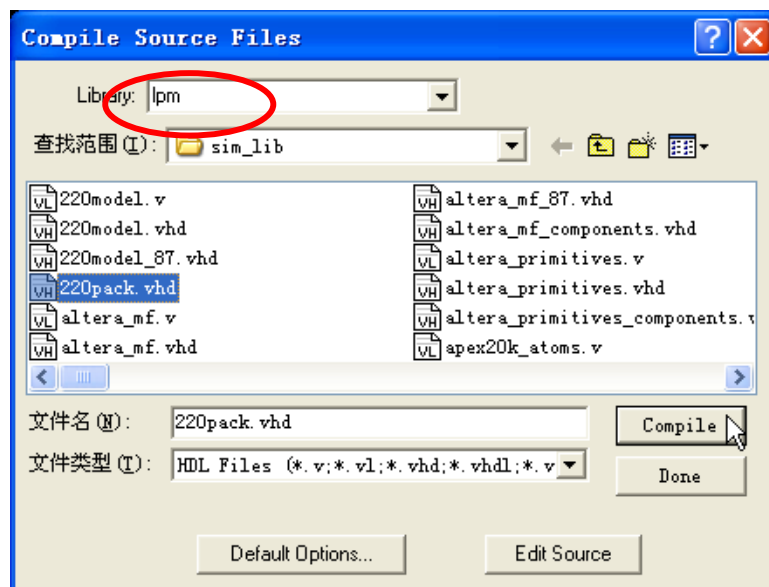
```
LIBRARY IEEE, cycloneii;
USE IEEE.std_logic_1164.all;
USE cycloneii.cycloneii_components.all;
```

USE cycloneii.cycloneii_components.all 这句就是调用 cycloneii 库里的 cycloneii_components 的意思。

5、将 lpm 仿真原型文件编译到 lpm 库中去。Comile→Compile...



- 6、点 **Library** 里面选中 **lpm**; 查找范围内找到 **QuartusII** 安装目录下...\eda\sim_lib 里面的仿真原型文件;



对 Verilog 和 VHDL, 分别选以下的文件:

- **220model.v** (Verilog HDL) [注意: 如果建Verilog的库最好在后面加上_ver, 例如lpm库, 建Verilog的要用lpm_ver]
- **220pack.vhd** and **220model.vhd** (VHDL)

注意, VHDL 建库时要注意先后次序, 先选中 **220pack.vhd**, 点 **Compile**, 再选中 **220model.vhd**, 点 **Compile**, 最后点 **Done**。这样, **lpm** 库就建好了。

- 7、下面按同样的方法把其它库也建好。

altera_mf (Verilog 的库名用 altera_mf_ver)

- **altera_mf.v** (for Verilog HDL)
- 先 **altera_mf_components.vhd** 后 **altera_mf.vhd** and (for VHDL) (注意先后顺序)

primitive (Verilog的库名用primitive_ver)

- **altera_primitives.v** (for Verilog HDL)
- 先**altera_primitives_components.vhd** 后**altera_primitives.vhd** (for VHDL) (注意先后顺序)

cycloneii (verilog的库名用cycloneii_ver. 以下XXX代表器件型号, 如cycloneii. 我们只需要添加常用的器件就行了, 根据设计添加元件库)

- **xxx_atoms.v** (for Verilog HDL)
- 先**xxx_atoms.vhd**后**xxx_components.vhd** (for VHDL) (注意先后顺序)

下面修改 modelsim.ini 文件

8、modelsim.ini 文件位于 ModelSim 的安装目录下。默认是一个只读文件。为保险起见，将原 modelsim.ini 文件先保留一份备份。

9、选中 modelsim.ini 文件点右键，去掉只读。

10、打开 modelsim.ini 文件，在[Library]和[vcom]之间加上以下类似语句，前面的 \$MODEL_TECH/..是指 ModelSim 的安装目录，如果 altera_lib 不是安装在 ModelSim 的安装目录下的，也可以直接写成 C:\altera_lib\VHDL\lpm 这样的形式。

.....

```
lpm = $MODEL_TECH/../../altera_lib/VHDL/lpm
```

```
altera_mf = $MODEL_TECH/../../altera_lib/VHDL/altera_mf
```

```
cyclone = $MODEL_TECH/../../altera_lib/VHDL/cyclone
```

```
cycloneii = $MODEL_TECH/../../altera_lib/VHDL/cycloneii
```

```
maxii = $MODEL_TECH/../../altera_lib/VHDL/maxii
```

```
stratixii = $MODEL_TECH/../../altera_lib/VHDL/stratixii
```

```
lpm_ver = $MODEL_TECH/../../altera_lib/Verilog/lpm_ver
```

```
altera_mf_ver = $MODEL_TECH/../../altera_lib/Verilog/altera_mf_ver
```

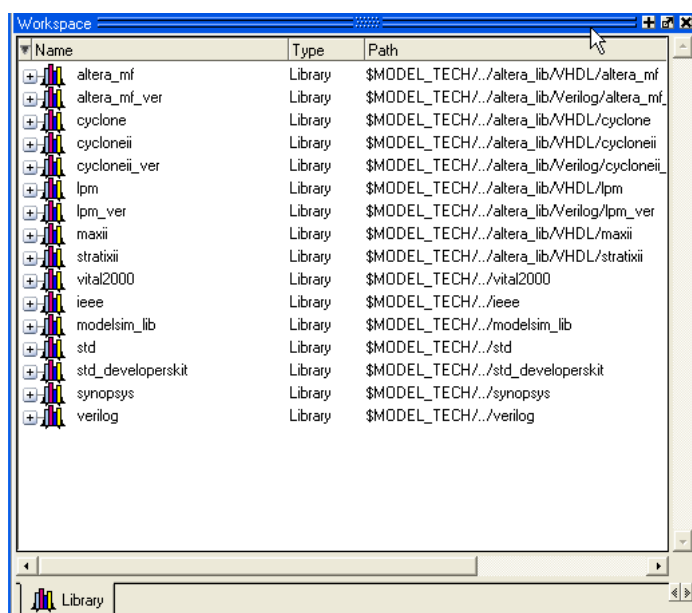
```
cycloneii_ver = $MODEL_TECH/../../altera_lib/Verilog/cycloneii_ver
```

.....

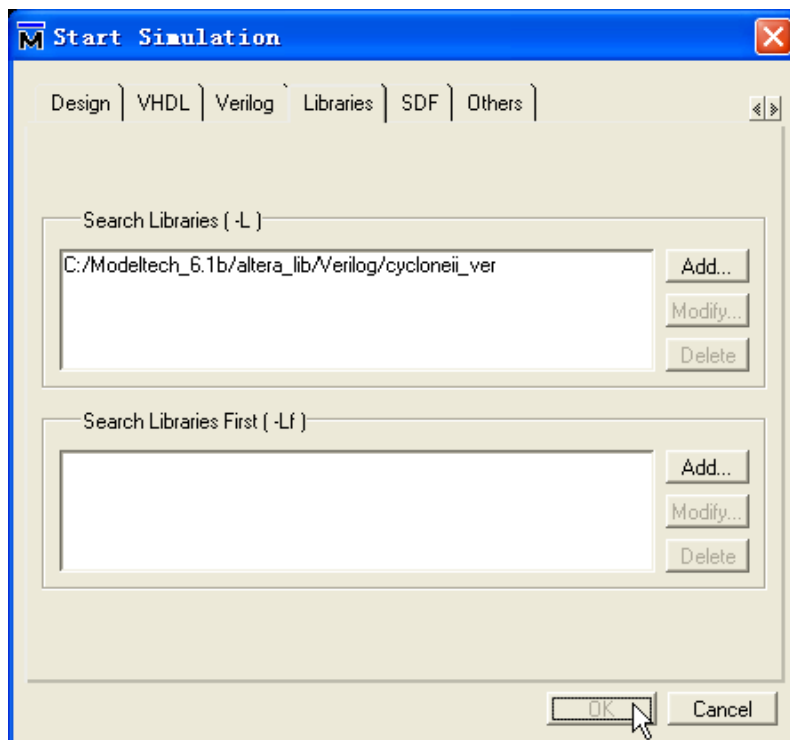
11、点保存，退出，选中 modelsim.ini，点右键，再把 modelsim.ini 重新设置为只读文件。

12、OK，至此，仿真库就完全建好了。

建好后，ModelSim 指向任何一个目录下或新建任何一个工程（project）时，ALTERA 的仿真库都会出现在资源库当中：



13、在仿真时，可以指定 ALTERA 仿真库的位置。例如：



有关于功能仿真和时序仿真的步骤已记录于《用 ModelSimSE 进行功能仿真和时序仿真的方法（ALTERA 篇）》之中。

（完）

ModelSim 学习笔记(三)

用 ModelSimSE 进行
功能仿真和时序仿真的方法
(ALTERA 篇)

黄俊

April 2007

以前用的是 LATTICE 的，ispLEVER 有自带了一个 OEM 版的 ModelSim。要仿真时，不需要添加库，用起来比较方便，自己有点懒，所以就一直凑合着用。现在转向用 ALTERA 了，ALTERA 也有 OEM 版的 ModelSim，也不用添加库。后来听说 ModelSim SE 的功能更强大，速度更快，所以就决定把 ModelSim SE 好好摸索一下，再多学习一点关于 TestBench 技巧方面的知识。

我的学习资料主要是 ModelSim SE 自带的教程、ALTERA 提供的资料以及 edacn 上面 ModelSim 专栏由网友们上传的资料。因为是初学，加上看到英文资料一大堆，烦都烦死，而有些中文文档可能是有些步骤没有讲清楚，我实际按照文档上面说的一步一步做下来也老是完成不了，花了不少时间。我于是就想自己摸清楚后，把步骤截图下来，整理清楚，做成笔记。一方面加深自己的认识，另一方面对初学者也许会有些许用处。

我近期计划陆续整理出以下几个方面的学习笔记：

- 初学 ModelSimSE 时被迷糊了几天的若干概念
- 在 ModelSimSE 中添加 ALTERA 仿真库的详细步骤
- 用 ModelSimSE 进行功能仿真和时序仿真的方法（ALTERA 篇）
- ModelSimSE 中常用到的几个命令及 DO 文件的学习笔记
- 近来学到的几招 TestBench 的技巧

MSN: paulhuang_sz@hotmail.com

E-mail: huangjun5927@163.com

Blog: <http://www.edacn.net/index.php/2599>

用 ModelSim SE 进行功能仿真和时序仿真的方法

(ALTERA 篇)

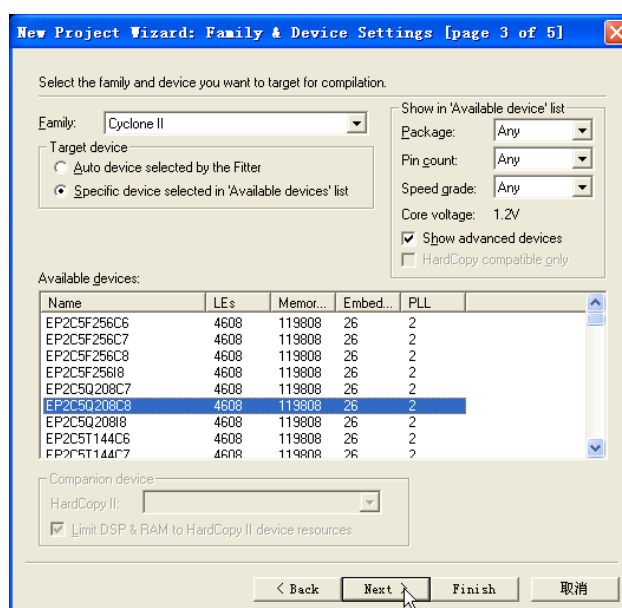
软件准备

- (1) QuartusII, 本文截图是 QuartusII 6.1 界面的。我个人认为, 如果是开发 StratixII 或 CycloneII 或 MAXII, 用 QuartusII6.0+SP1+SP2 比较稳定。
- (2) ModelSim SE. ALTERA 仿真库要已经装好, 安装仿真库的笔记已记录于《在 ModelSimSE 中添加 ALTERA 仿真库的详细步骤》中。我电脑上装的是 ModelSim SE 6.1b。

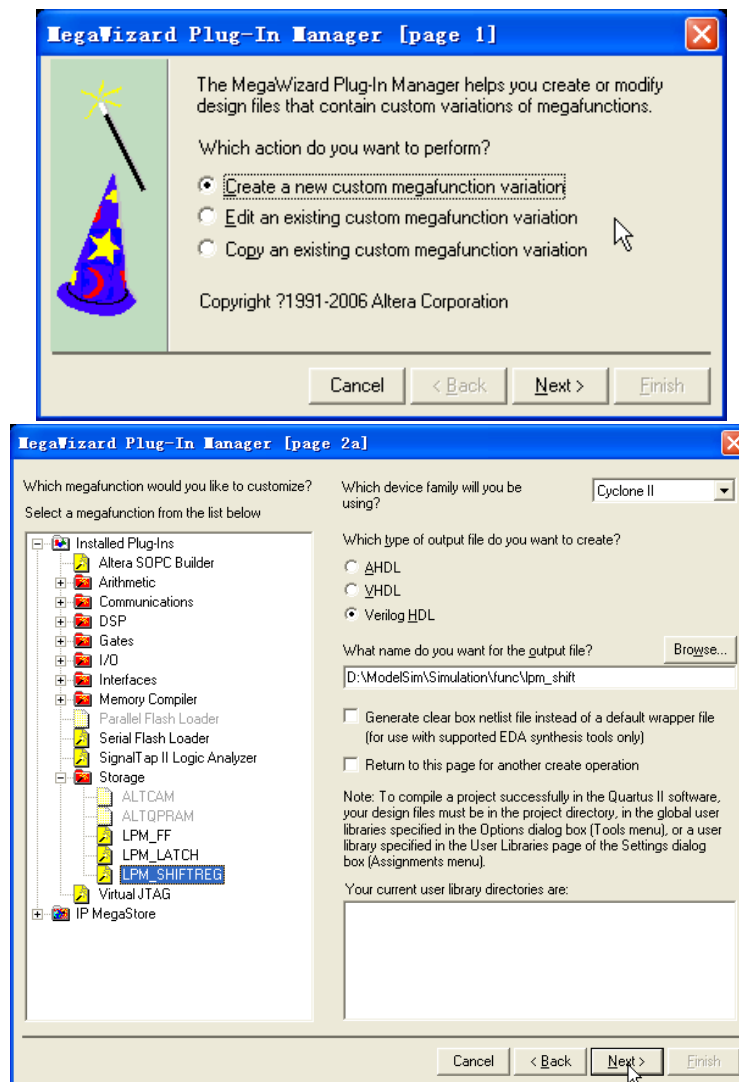
例子程序的制作

先在 Quartus II 里生成一个例子程序, 以方便介绍三种仿真的方法。步骤如下:

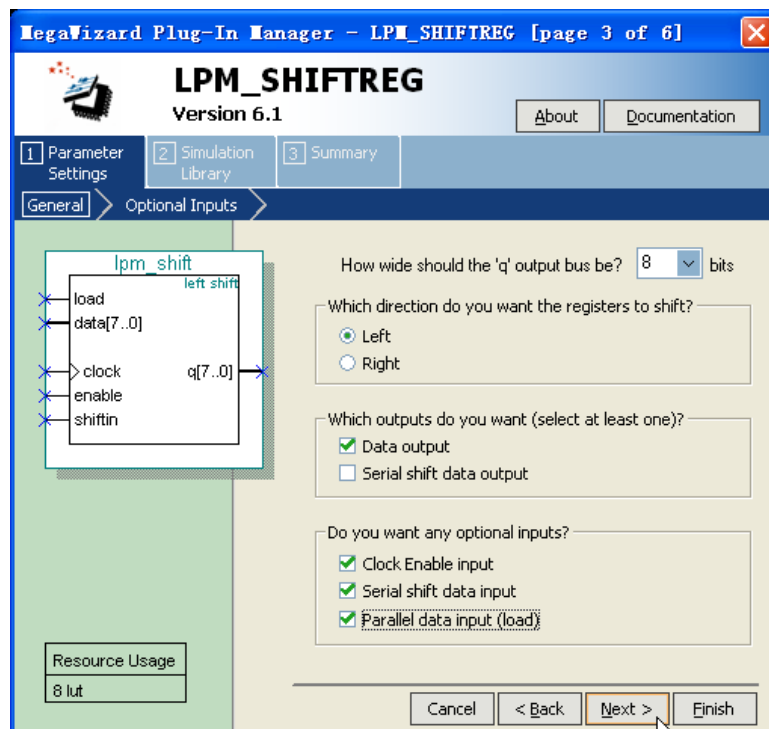
- 1、新建一个工程 (**Project**), 工程名取 **lpm_shift**, 器件选 **CycloneII EP2C8Q208C**, 第三方的工具暂时都不选。



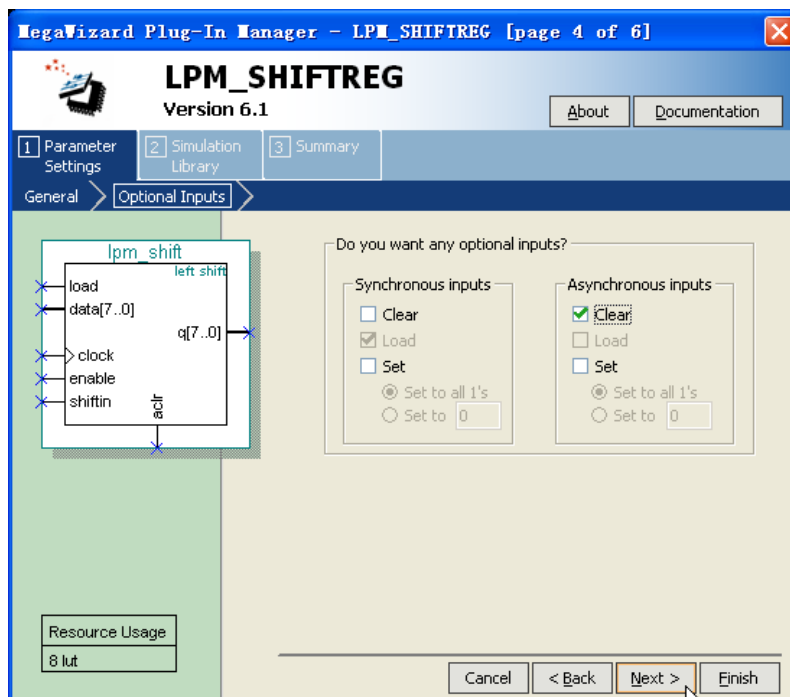
- 2、菜单栏上 **Tools**→**MegaWizard Plug-In Manager**, 点 **Next**, 在 **storage** 中选 **LPM_SHIFTREG**, 输出文件格式根据习惯选一种语言, 在这里以 **Verilog** 的为例, 在右边的 **output file** 名字中加上 **lpm_shift**. 点 **Next**。



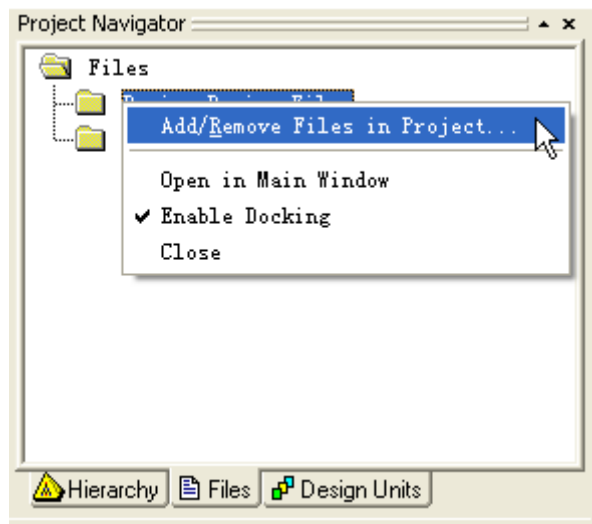
- 3、这个例子是做一个移位寄存器，调用 **lpm** 库，和 **cycloneII** 元件库，也正好可以作为对前面建好的 **ALTERA** 库的一个验证。点 [Documentation](#) 可以查到该模块的使用说明和详细介绍。移位寄存器比较简单，就不用细看了。如下图设置.点 **Next**.




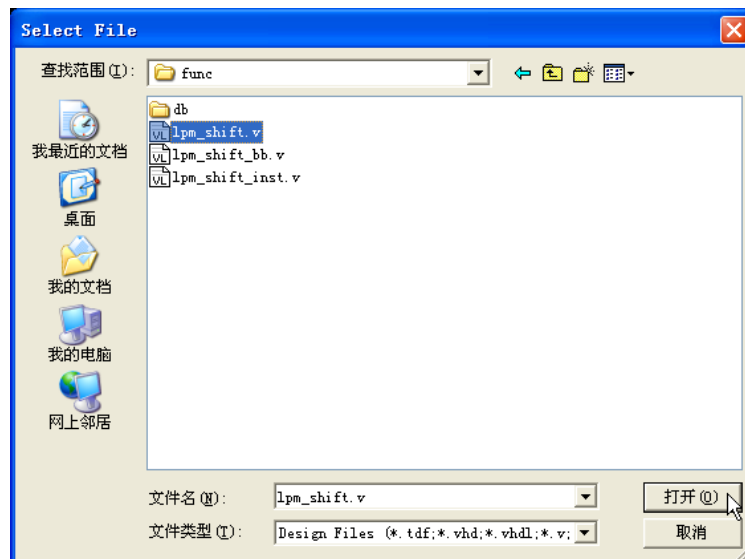
4、加上一个异步清零端，点 **Next**，再点 **Next**,最后点 **Finish**.



5、回到 **QuartusII** 主界面，点 **File** 选项卡，在 **Device Design File** 上面点右键，选择 **Add/Remove Files in Project...**,



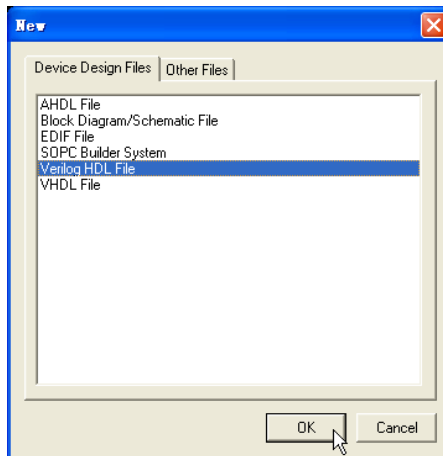
6、点图标，选中生成的 **lpm_shift**，点打开再点 **Add** 都加到项目中去。




7、在这里直接把 **lpm_shift.v** 当成顶层文件，就不需要再例化它了。

8、下面写一个简单的 **TestBench**。

a) Quartus II 工具栏上点图标，选 **Verilog HDL**;



- b) 点 , 设文件名为 **top_tb**;
- c) 如下所述写一个简单的 **TestBench**.(例子见附件)

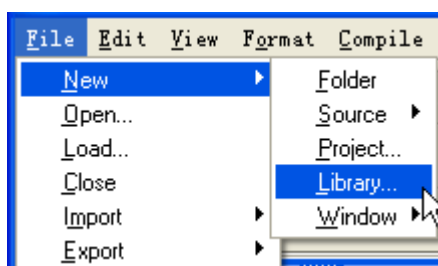
开始仿真

- **ModelSim** 仿真有很多种流程,下面我采用个人感觉比较好一个流程进行仿真实验验证:
基于工程 (Project) 的流程
 - Step1 新建一个工程
 - Step2 添加文件到工程中去
 - Step3 编译设计文件
 - Step4 启动仿真器, 指定顶层设计单元
 - Step5 查看和调试结果
- 对 ALTERA 的设计有三个阶段的仿真。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。

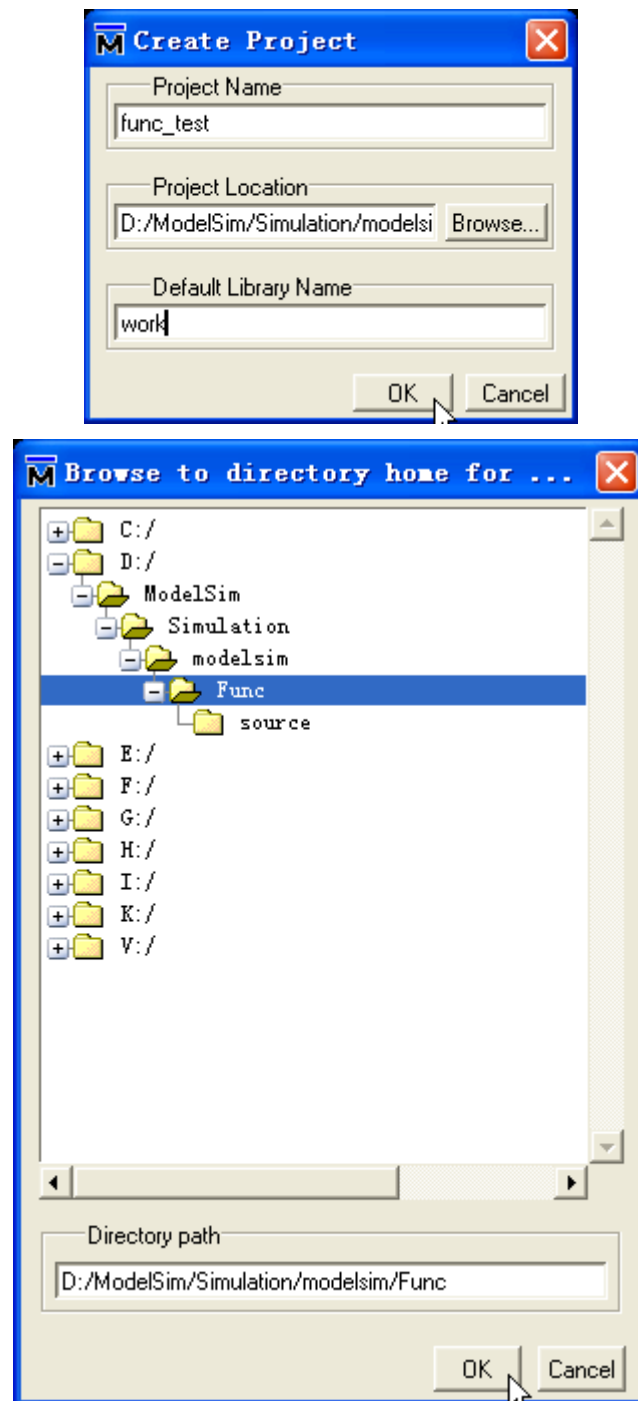
纯粹的功能仿真

1、新建一个工程

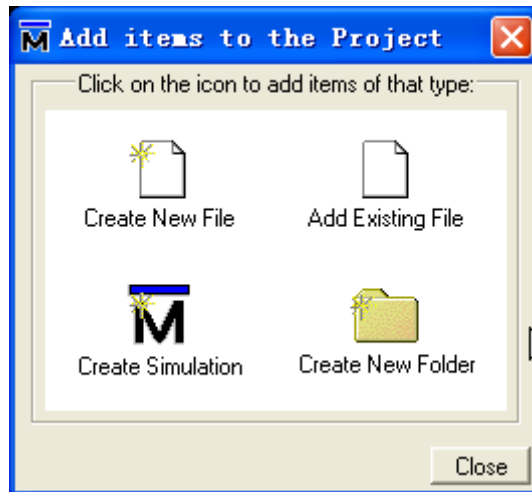
- a) 打开 **ModelSim SE**;
- b) 新建工程, **File→New→Library...**:



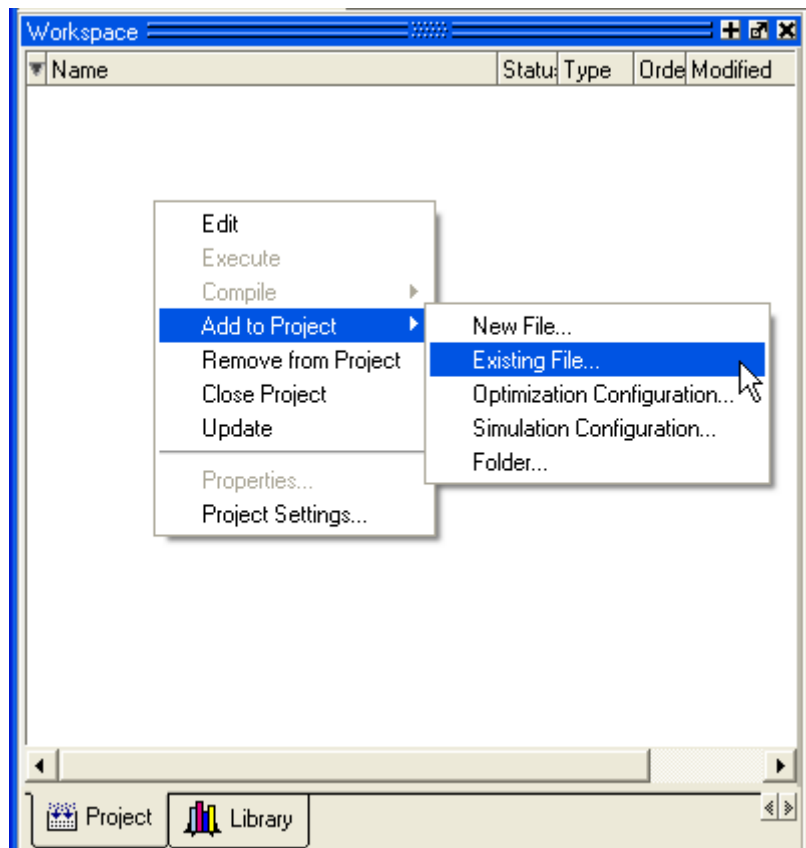
- c) 输入工程名，指定工程保存路径。为了可以分别清楚地对三种仿真进行验证，我建立了三个文件夹，**func**、**Psyth**、**PAR** 分别来当功能仿真，综合后功能仿真以及布局布线后时序仿真的工程保存文件夹。在这些文件夹内，我又建立了 **source** 文件夹，专门用来存放输入文件。将前面建立的 **lpm_shift.v** (HDL 源文件) 和 **top_tb.v** (Testbench 文件) 拷贝至 **func/source** 里面。将 **top_tb.v** 拷贝至 **Psyth**、**PAR** 里的 **source** 文件夹内。在这里先点 **Browse...**，将工程的保存路径指定到 **func** 文件夹内。



- d) 点 **Add Existing File**，添加输入文件。



e) 在 Project 的空白处，点右键也可以再添加存在的文件的方式添加输入文件。



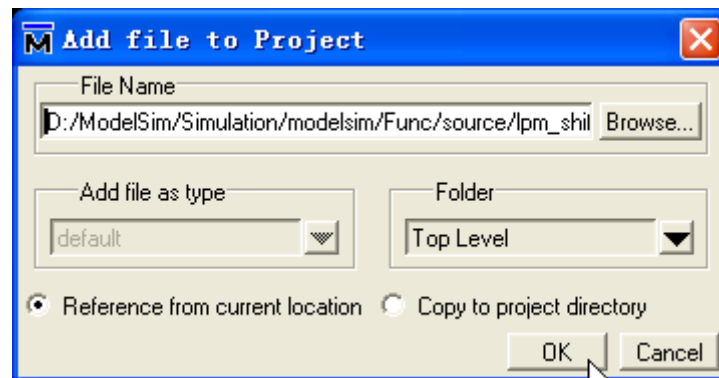
2、添加文件到工程中去

进行功能仿真需要的输入文件：

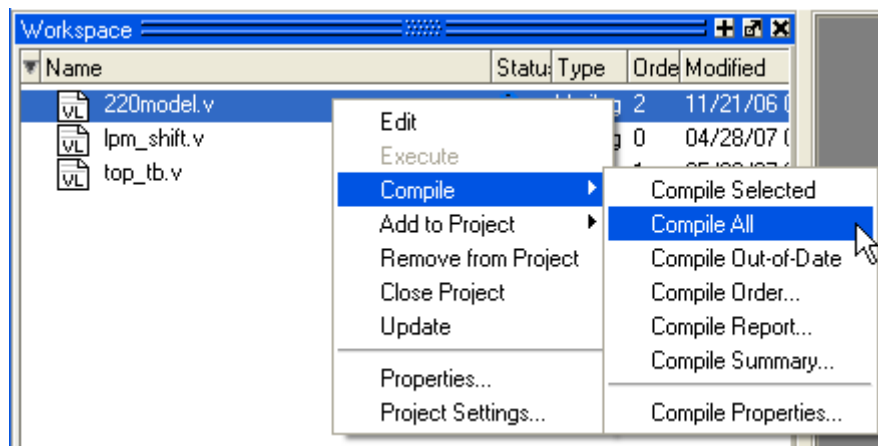
- HDL 文件；
- TestBench 文件；
- 仿真原型文件（在这时就添加进来）或预编译的库文件（仿真时指定）；

点 **Add Existing File** 添加 **lpm_shift.v**（HDL 源文件），**top_tb.v** (Testbench 文件)；再到 QuartusII 安装目录下 **..\altera\61\quartus\eda\sim_lib** 里面找到仿真原型文件 **220model.v**；用 **Reference from current location** 是链接的方式指定这些输入文件，**copy**

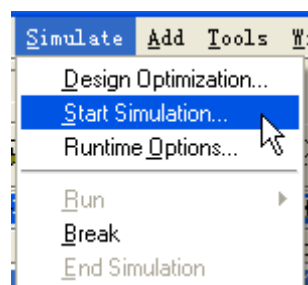
to project directory 是把输入文件复制到当前工程目录下。建议还是用 **Reference form current location**。



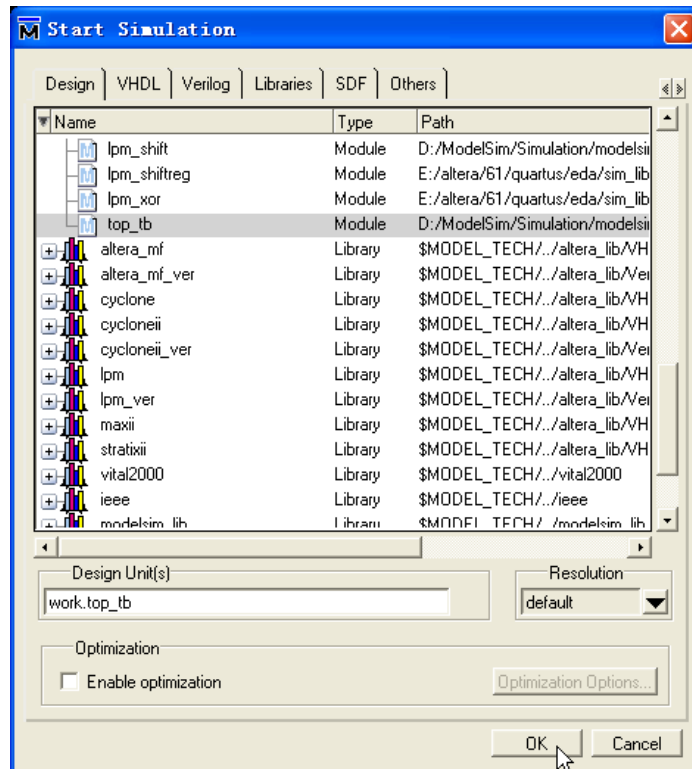
- 3、编译设计文件；
选中一个文件，点右键，选择 **Compile→Compile All**;



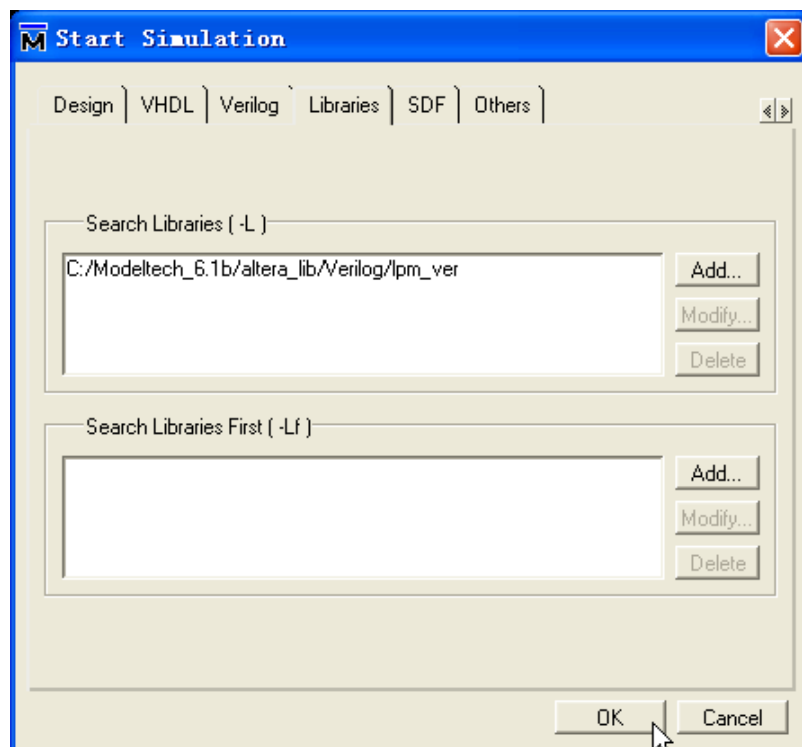
- 4、启动仿真器，指定顶层设计单元；
a) 进 **Simulate→Start Simulation...**



- b) 展开 **Work** 库，指定 **Testbench**，点 **OK** 开始仿真；

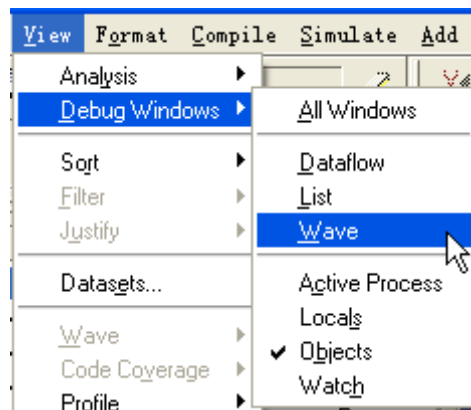


注：以上是直接添加了仿真原型文件的。如果已经预编译了库文件，就不需要这么麻烦，直接在仿真的时候指定就行了。

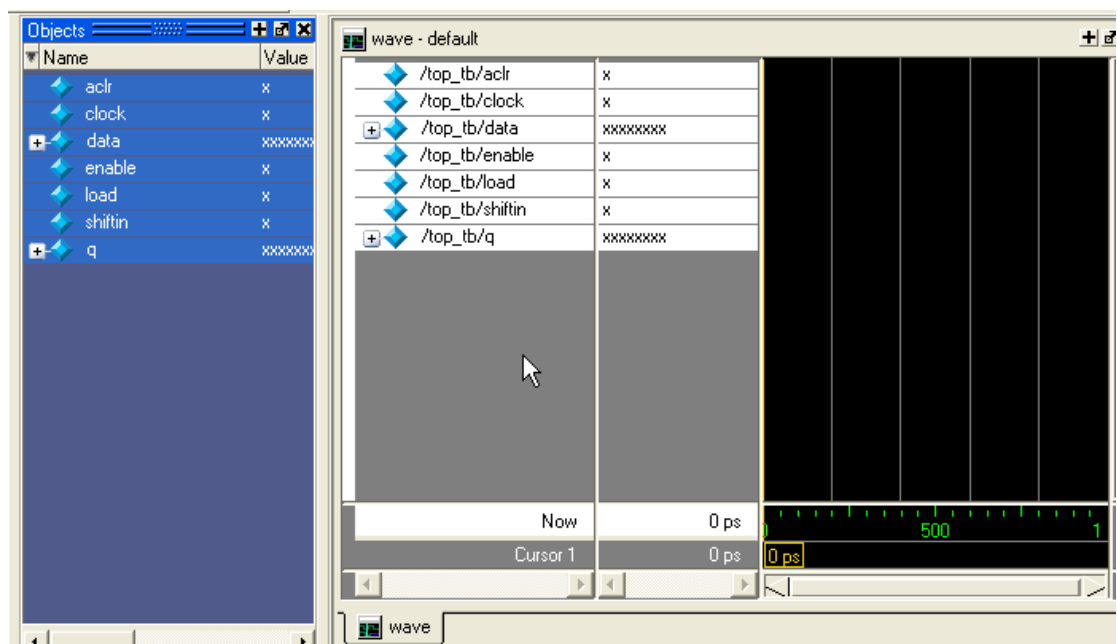


5、查看和调试结果

a) 将 **Wave** 窗口打开，查看仿真波形：




b) 将需要查看的信号拖入波形窗口，可以用 **Ctrl** 或 **Shift** 键多选。

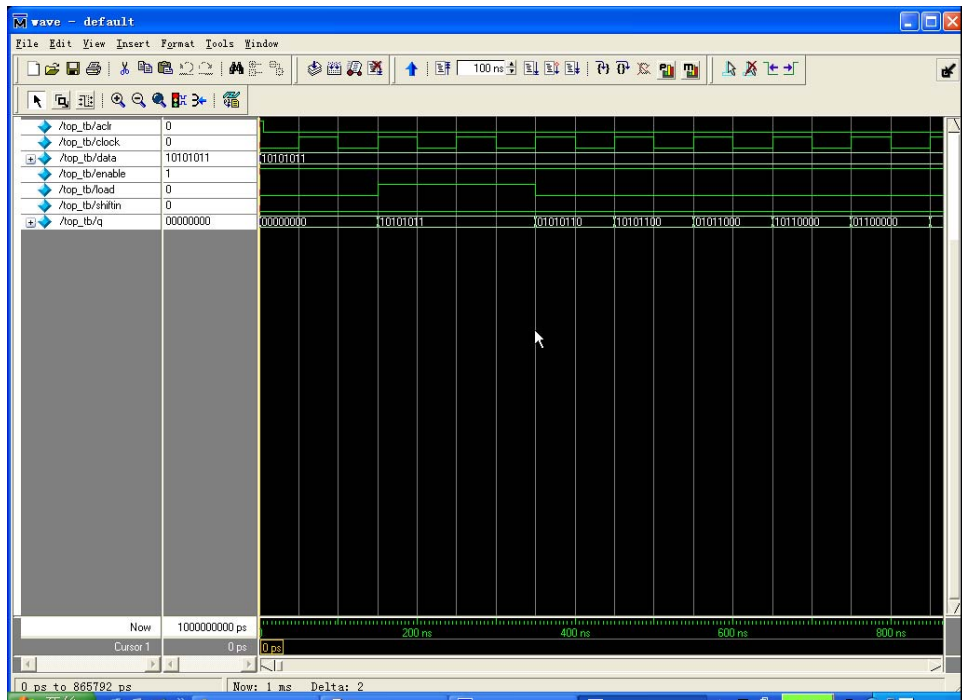



c) 运行。在下面的命令行中输入运行的时间，回车。

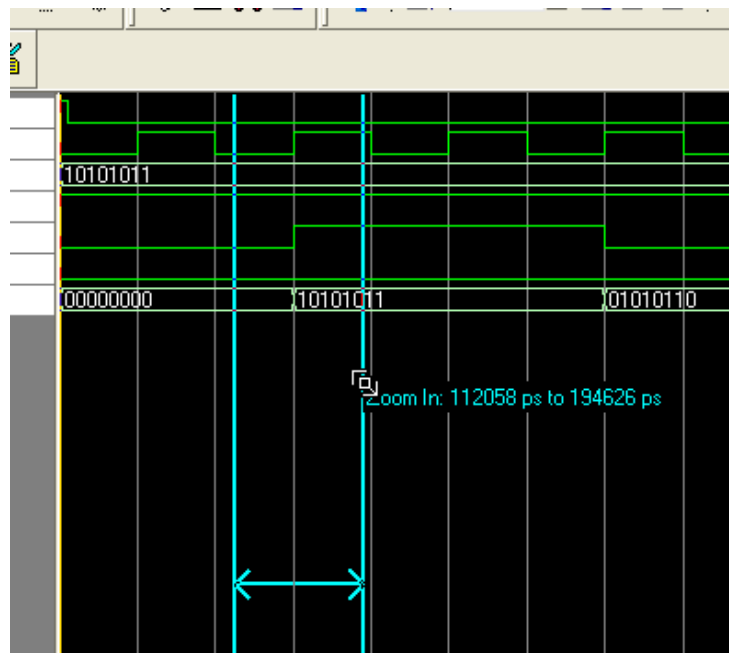
```
Transcript
ModelSim> vsim work.top_tb
# vsim work.top_tb
# Loading work.top_tb
# Loading work.lpm_shift
# Loading work.lpm_shiftreg
view wave
# .main_pane.mdi.interior.cs.vm.paneset.cli_0.wf.clip.cs.pw.wf

VSIM 13> run 1 ms
```

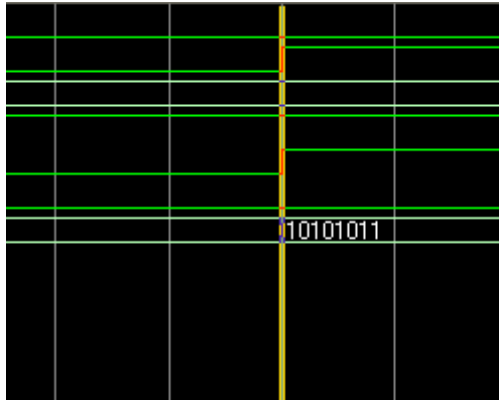
d) 查看结果：点  图标，最大化波形窗口。


















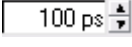





- e) 功能仿真的特点。点图标，在波形窗口下，按住左键不放，向右下斜拉可以选择一个放大的区域：



- f) 放大后可以看到，完全是没有延时的。这就是纯粹的功能仿真。



6、ModelSim 的其它使用技巧不在本文讨论范围之内，在这里就不介绍了。在这里只是列出几个常用按钮的作用。

- 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。

综合后功能仿真

1、新建一个工程

方法前面讲过了，这次将路径保存到.../modelsim/Pyth 里。

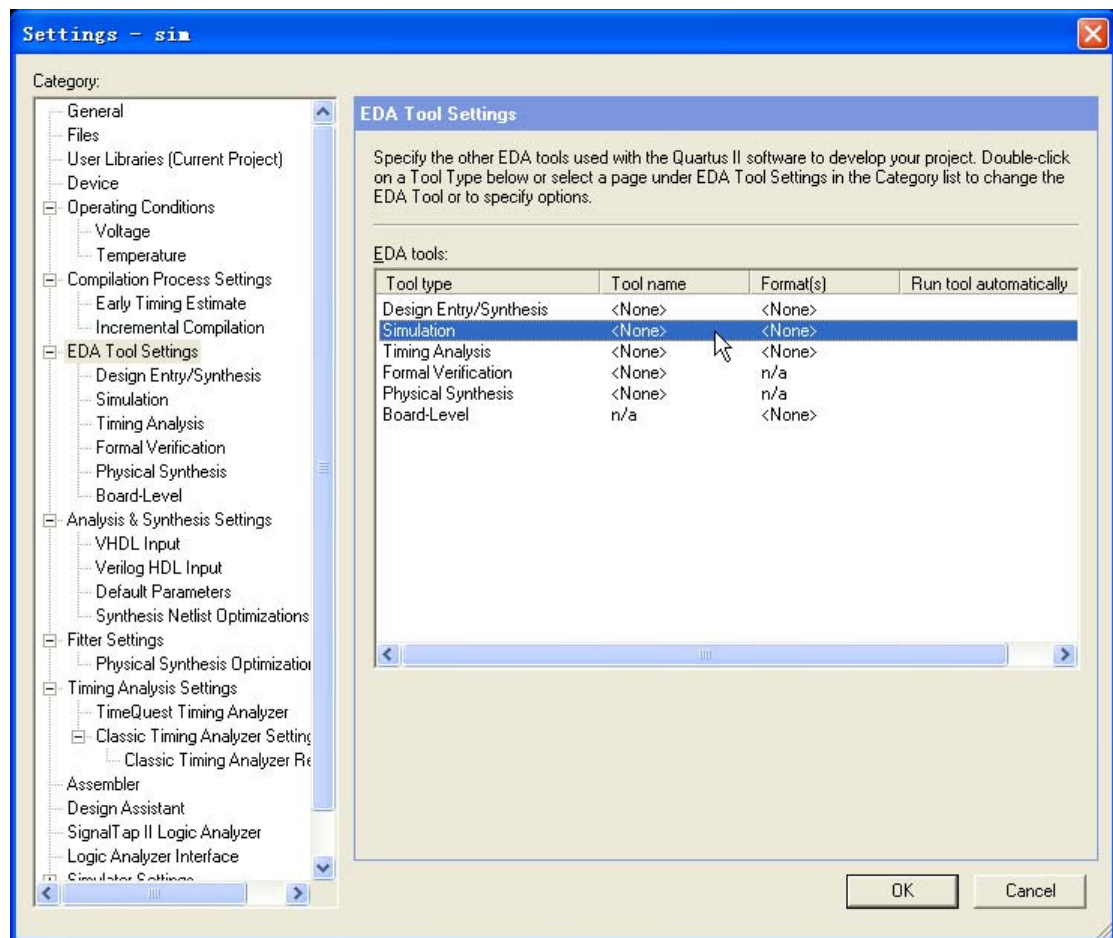
2、添加文件到工程中去

进行综合后功能仿真需要的输入文件：

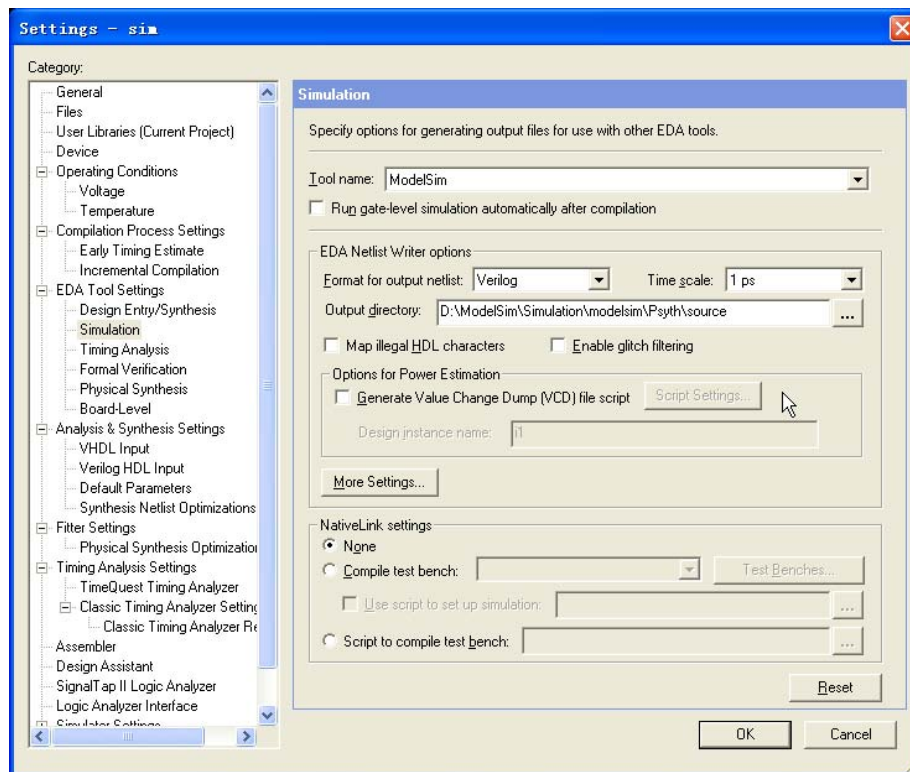
- 在 QuartusII 里面生成的网表文件；
- TestBench 文件；
- 仿真原型文件（在这时就添加进来）或预编译的库文件（仿真时指定）。

在 QuartusII 里面生成网表文件的方法：

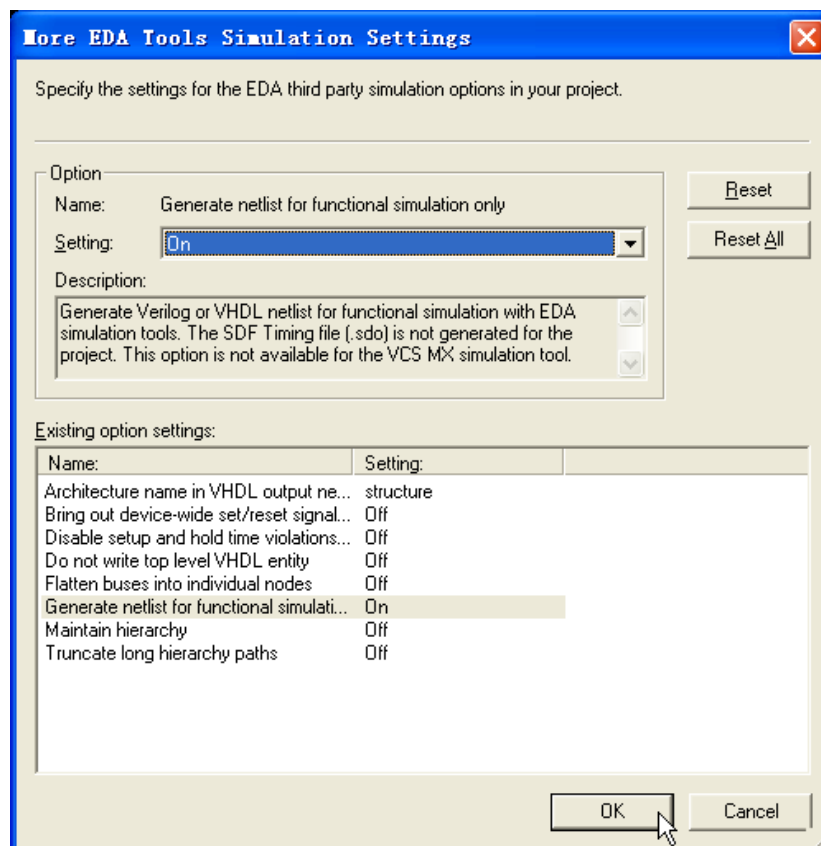
- a) 点工具栏上的  图标，选 **EDA Tool Setting**, 双击 **Simulation**。




- b) 设置 ModelSim, 输出网表的格式, 以及网表文件保存的路径。

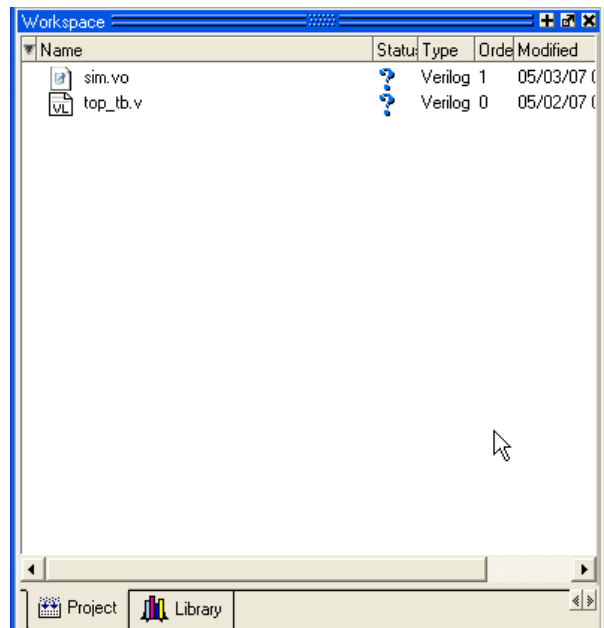


- c) 点 More Settings...，将 **Generate netlist for functional simulation only** 设置为 ON。

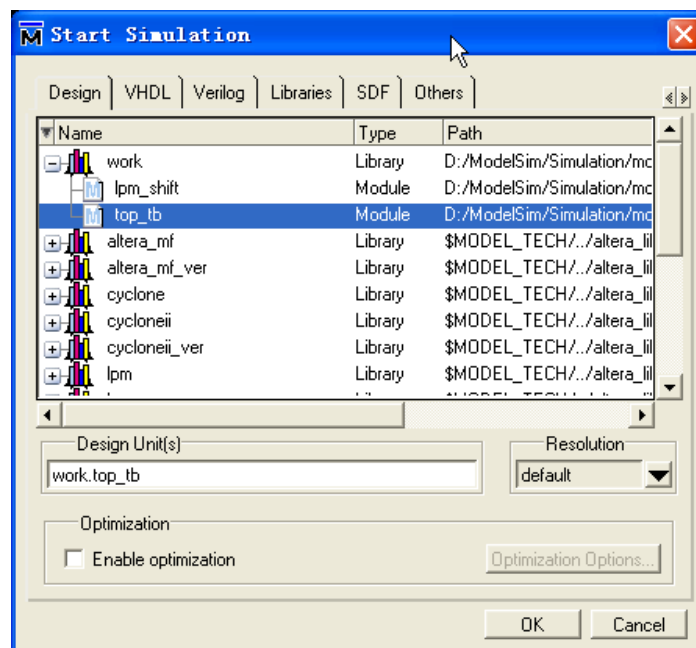


- d) 点 OK，保存。点工具栏上的  图标，进行全编译。

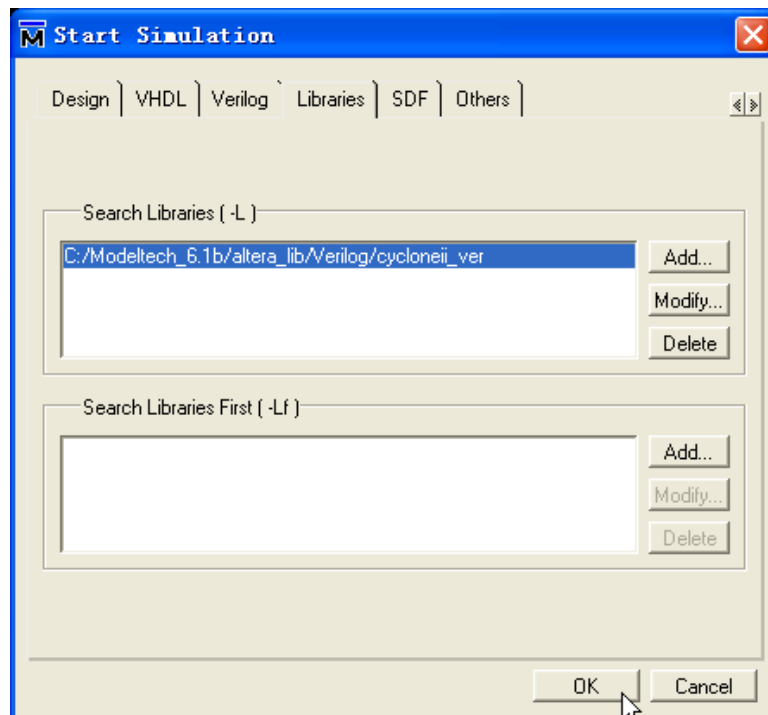
- e) 检查一下.../modelsim/Pyth/source 里面，可以发现，已经生成了一个*.vo 文件。
这个就是需要的网表文件。（VHDL 的输出网表是*.vho 后缀名的）
- f) 在这里，TestBench 文件就继续用前面功能仿真里用的 top_tb.v 文件；
- g) 在这里就采用直接添加仿真原型文件的方法了，到仿真时直接指定预编译的库。



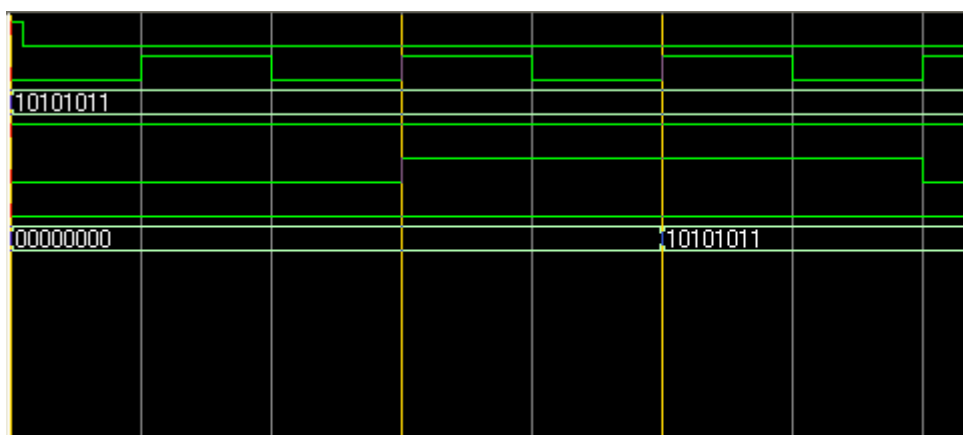
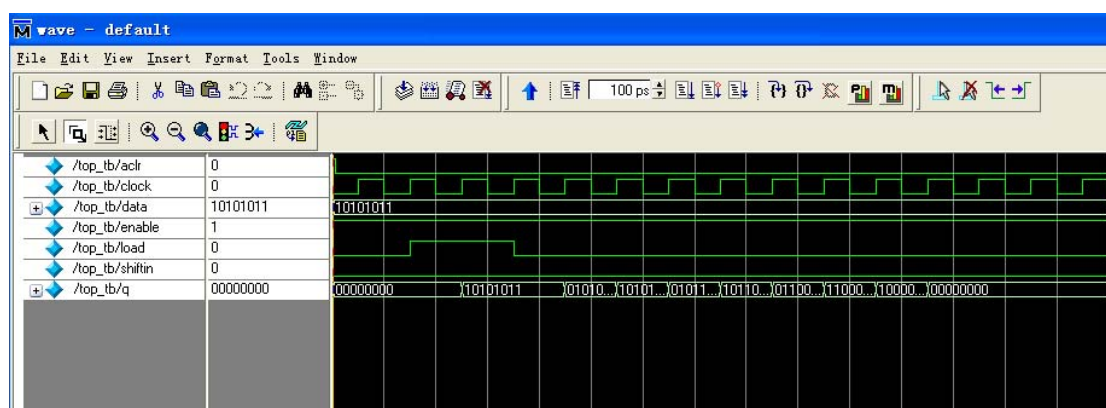
- 3、编译设计文件
(略)
- 4、启动仿真器，指定顶层设计单元
方法就不重复了。指定时参考以下截图。



这次选库的时候要选元件库了。



5、查看和调试结果



可以看到，load 在装载时，输出 q 延时一个时钟周期才输出，但它和输入时钟的边沿还是完全同步的。

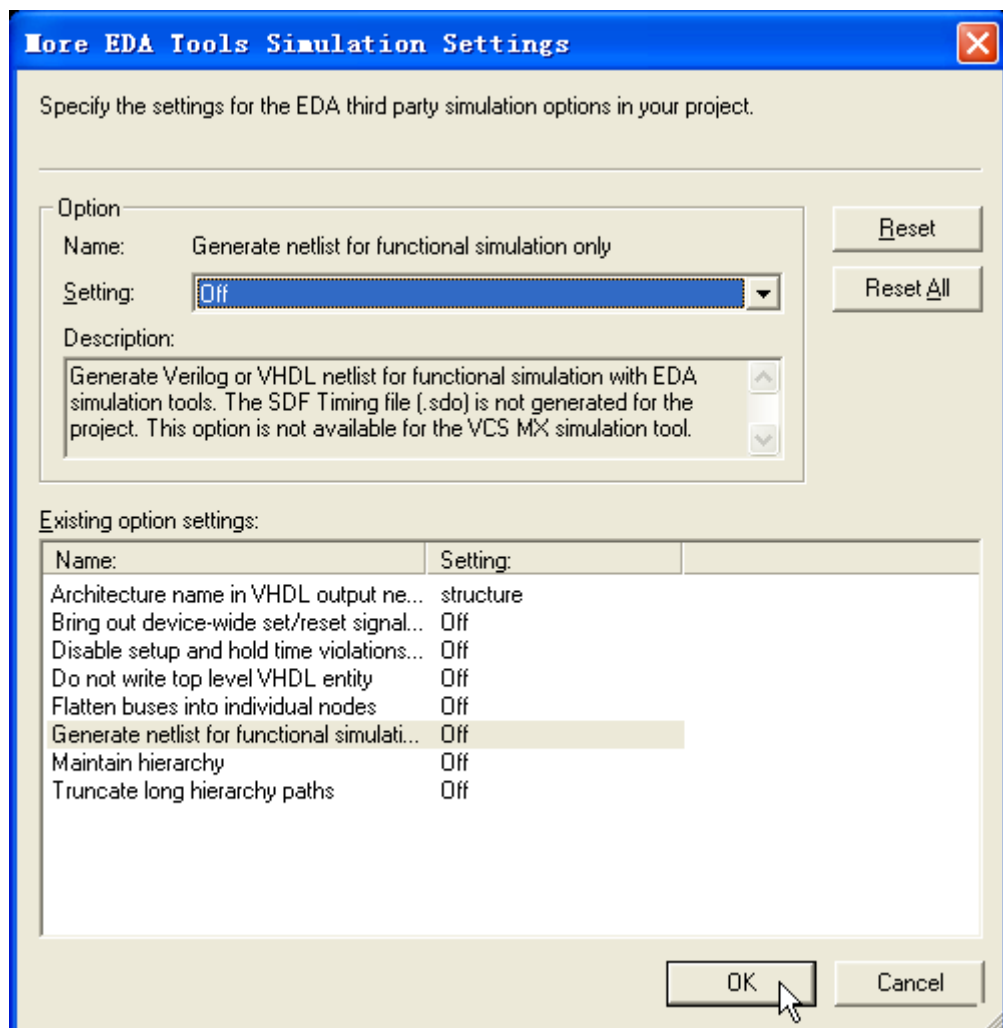
布局布线后时序仿真

步骤和综合后功能仿真大同小异，有以下几点要注意：

(1) 进行布线后时序仿真需要的输入文件：

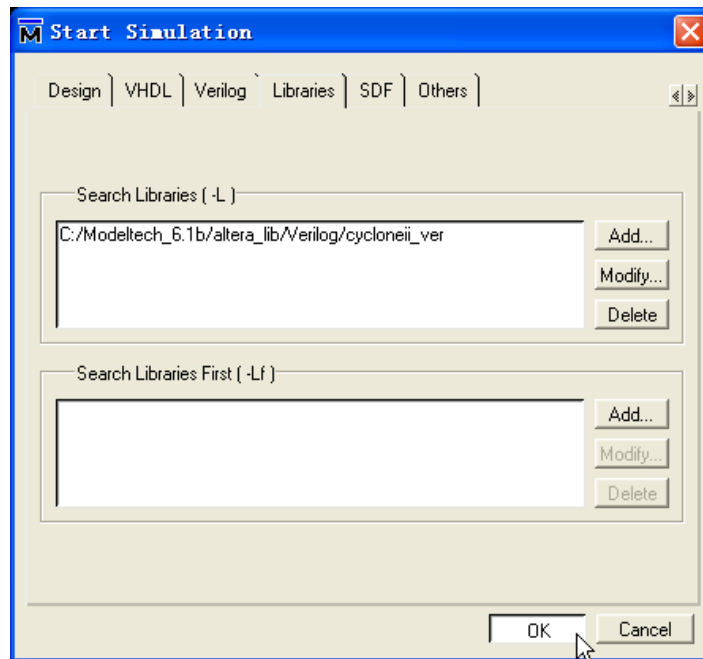
- 布局布线后网表；
- 延时文件*.sdo(Verilog)或*_yhd.sdo(VHDL)；
- TestBench 文件；
- 仿真原型文件(在这时就添加进来)或预编译的库文件(仿真时指定)；

QuartusII 里面设置重新把 **Generate netlist for functional simulation only** 设置成 OFF。

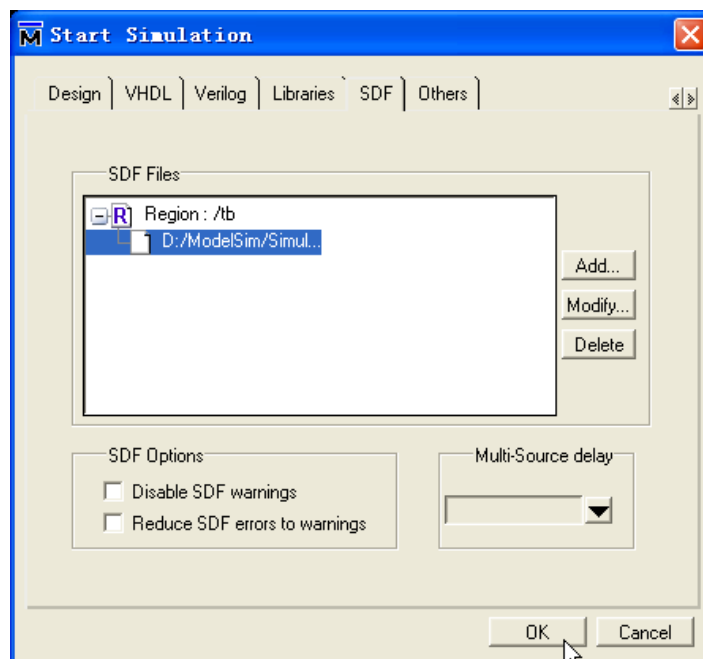


全编译后，除了生成布局布线后网表文件以外，还会生成延时文件。

(2) 依然要指定库文件的位置；

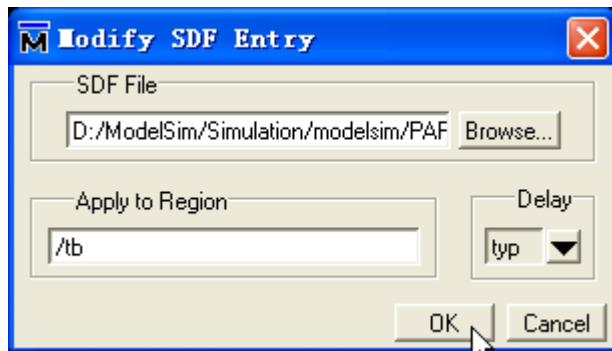


(3) 指定延时文件时，在下图位置指定延时文件。



(4) 需要注意两点：

- a) 对 Verilog 设计的仿真，延时文件需要复制到 ModelSim 里面建的 Project 的根目录下。这一点，我也搞不明白，试了很久才发现的，可能是我没有设置好的问题，也可能是这个版本本身的 Bug。对 VHDL 设计，就不需要复制到 ModelSim 里面建的 Project 的根目录下。
- b) 指定 Apply to Region 里面，还要写 TestBench 里面例化顶层文件的例化名。

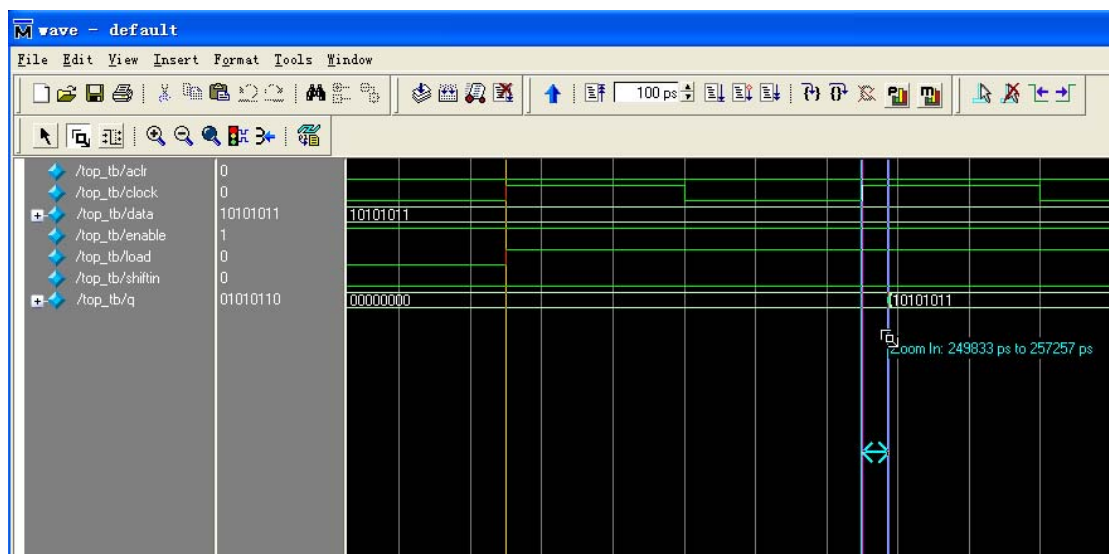


```

topm shift tb (
    .aclr ( aclr ),
    .clock (clock),
    .data (data),
    .enable ( enable ),
    .load (load),
    .shiftin ( shiftin ),
    .q ( q )
);

```

(5) 查看仿真波形，q 的输出就显示了延时。



(完)