

# Modelsim 6.0 使用教程

## 1. Modelsim 简介

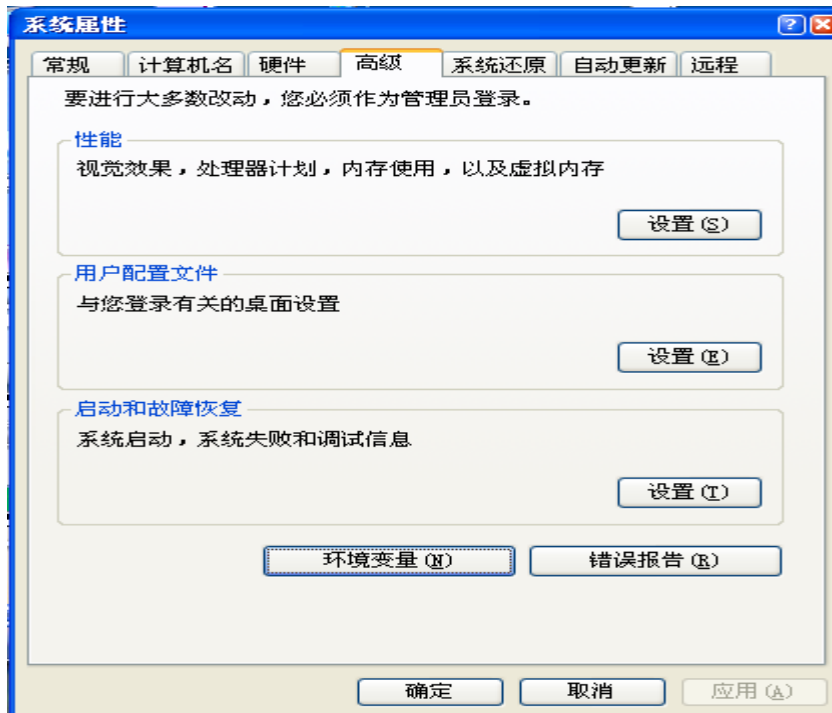
Modelsim 仿真工具是 Model 公司开发的。它支持 Verilog、VHDL 以及他们的混合仿真，它可以将整个程序分步执行，使设计者直接看到他的程序下一步要执行的语句，而且在程序执行的任何步骤任何时刻都可以查看任意变量的当前值，可以在 Dataflow 窗口查看某一单元或模块的输入输出的连续变化等，比 quartus 自带的仿真器功能强大的多，是目前业界最通用的仿真器之一。

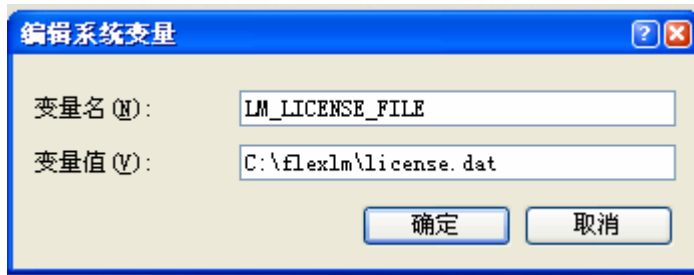
对于初学者，modelsim 自带的教程是一个很好的选择，在 Help->SE PDF Documentation->Tutorial 里面。它从简单到复杂、从低级到高级详细地讲述了 modelsim 的各项功能的使用，简单易懂。但是它也有缺点，就是它里面所有事例的初期准备工作都已经放在 example 文件夹里，直接将它们添加到 modelsim 就可以用，它假设使用者对当前操作的前期准备工作都已经很熟悉，所以初学者往往不知道如何做当前操作的前期准备。

## 2. 安装

同许多其他软件一样，Modelsim SE 同样需要合法的 License，通常我们用 Keygen 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 盘根目录新建一个文件夹 flexlm，用 Keygen 产生一个 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，变量值内如果已经有别的路径了，请用“;”将其与要填的路径分开。LM\_LICENSE\_FILE = c:\flexlm\license.dat





(4). 安装完毕，可以运行。

### 3. Modelsim 仿真方法

Modelsim 的仿真分为前仿真和后仿真，下面先具体介绍一下两者的区别。

#### 3.1 前仿真

前仿真也称为功能仿真，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.

#### 3.2 后仿真

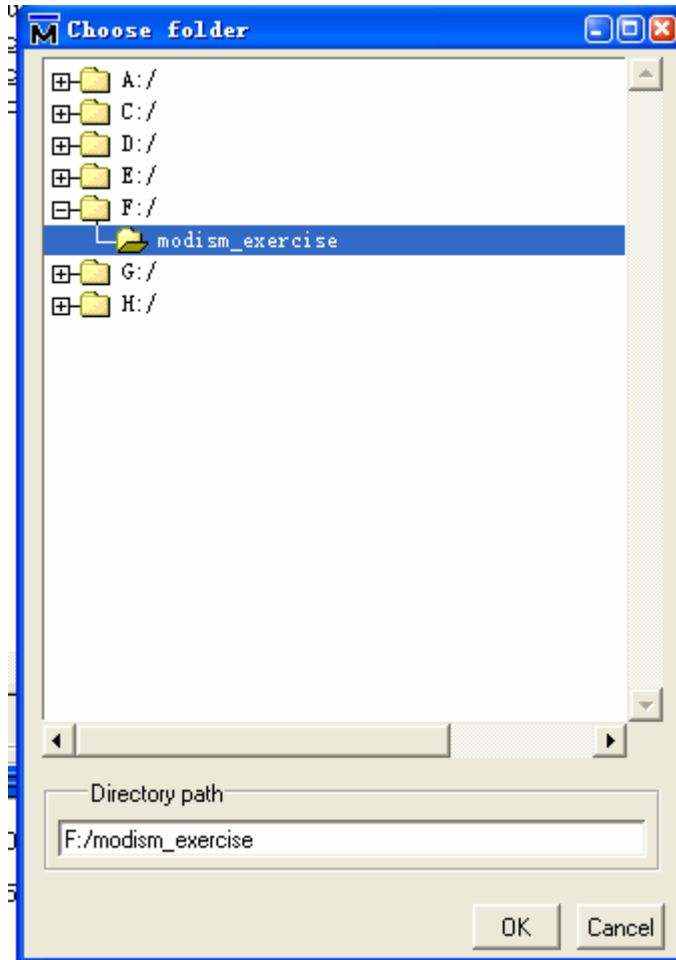
后仿真也称为时序仿真或者布局布线后仿真，是指电路已经映射到特定的工艺环境以后，综合考虑电路的路径延迟与门延迟的影响，验证电路能否在一定时序条件下满足设计构想的过程，是否存在时序违规。其输入文件为从布局布线结果中抽象出来的门级网表、Testbench 和扩展名为 SDO 或 SDF 的标准时延文件。SDO 或 SDF 的标准时延文件不仅包含门延迟，还包括实际布线延迟，能较好地反映芯片的实际工作情况。一般来说后仿真是必选的，检查设计时序与实际的 FPGA 运行情况是否一致，确保设计的可靠性和稳定性。

#### 3.3 Modelsim 仿真的基本步骤

Modelsim 的仿真主要有以下几个步骤：建立库并映射库到物理目录；编译原代码（包括 Testbench；执行仿真。

##### 3.3.1 建立库

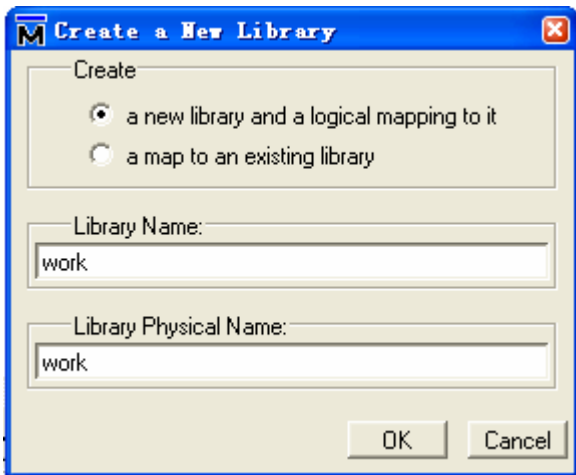
在执行一个仿真前先建立一个单独的文件夹，后面的操作都在此文件下进行，以防止文件间的误操作。然后启动 Modelsim 将当前路径修改到该文件夹下，修改的方法是点 File->Change Directory 选择刚刚新建的文件夹见下图。



仿真库是存储已编译设计单元的目录，modelsim 中有两类仿真库，一种是工作库，默认的库名为 work, 另一种是资源库。Work 库下包含当前工程下所有已经编译过的文件。所以编译前一定要建一个 work 库，而且只能建一个 work 库。资源库存放 work 库中已经编译文件所要调用的资源，这样的资源可能有很多，它们被放在不同的资源库内。例如想要对综合在 cyclone 芯片中的设计做后仿真，就需要有一个名为 cyclone\_ver 的资源库。

映射库用于将已经预编译好的文件所在的目录映射为一个 modelsim 可识别的库，库内的文件应该是已经编译过的，在 Workspace 窗口内展开该库应该能看见这些文件，如果是没有编译过的文件在库内是看不见的。

建立仿真库的方法有两种。一种是在用户界面模式下，点 File->New->Library 出现下面的对话框，选择 a new library and a logical mapping to it, 在 Library Name 内输入




要创建库的名称，然后 OK，即可生成一个已经映射的新库。另一种方法是在 Transcript 窗口输入以下命令：

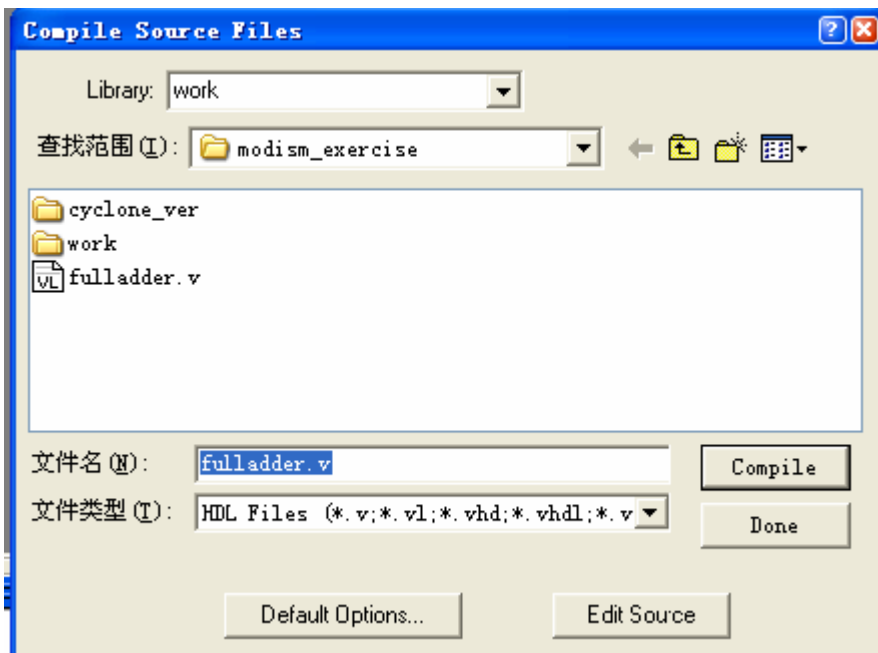
```
vlib          work
             /*      库名      */
vmap          work          work
             /*      映射的逻辑名称      存放的物理路径 */
```

如果要删除某库，只需选中该库名，点右键选择 Delete 即可。

需要注意的是不要在 modelsim 外部的系统盘内手动创建库或者添加文件到库里；也不要 modelsim 用到的路径名或文件名中使用汉字，因为 modelsim 可能无法识别汉字而导致莫名其妙的错误。

### 3.3.2 编写与编译测试文件

在编写 Testbench 之前最好先将要仿真的目标文件编译到工作库中，点 Compile->Compile 或 ，将出现下面的对话框，

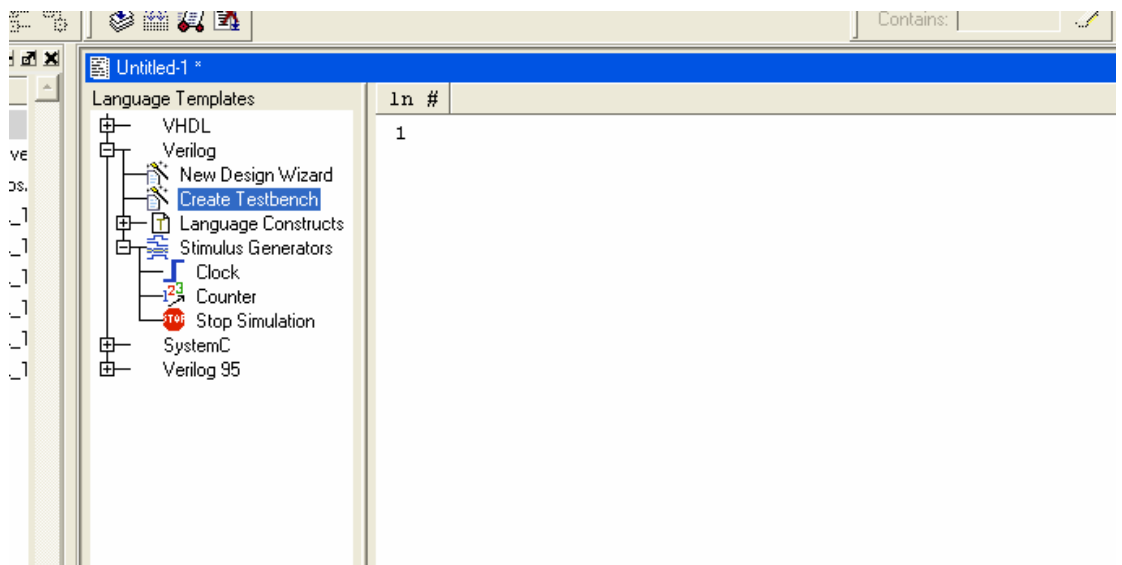


在 Library 中选择工作库，在查找范围内找到要仿真的目标文件，然后点 Compile 和 Done。或在命令行输入 vlog fulladder.v。此时目标文件已经编译到工作库中，在 Library 中展开工作库会发现该文件。

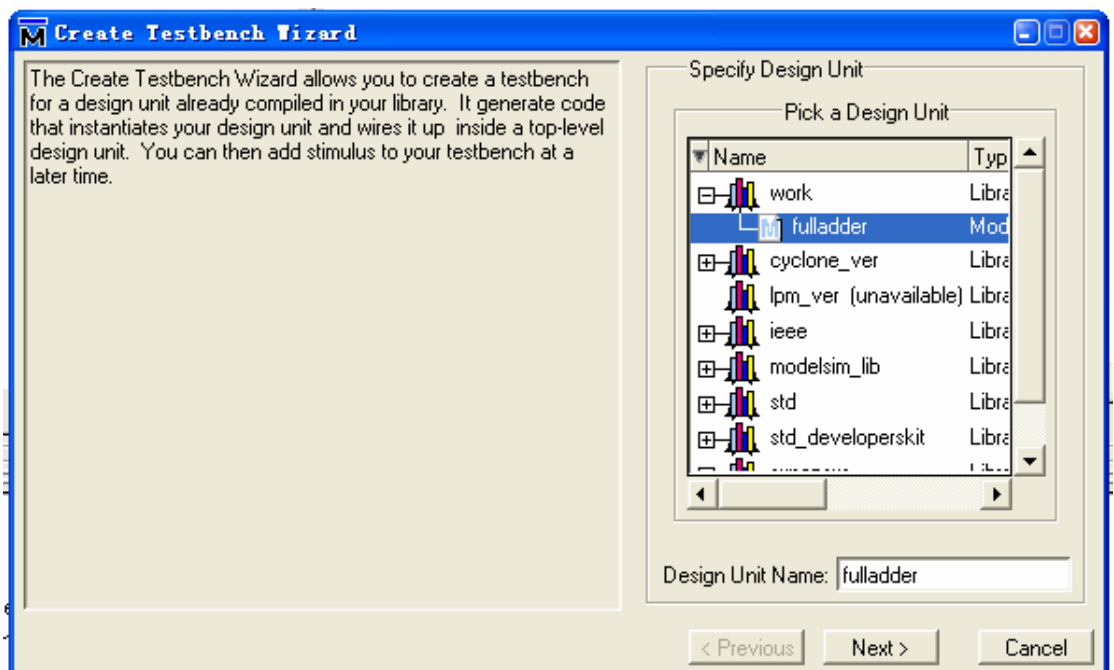
当对要仿真的目标文件进行仿真时需要给文件中的各个输入变量提供激励源,并对输入波形进行的严格定义,这种对激励源定义的文件称为 Testbench,即测试台文件。下面先讲一下 Testbench 的产生方法。

我们可以在 modelsim 内直接编写 Testbench,而且 modelsim 还提供了常用的各种模板。具体步骤如下:

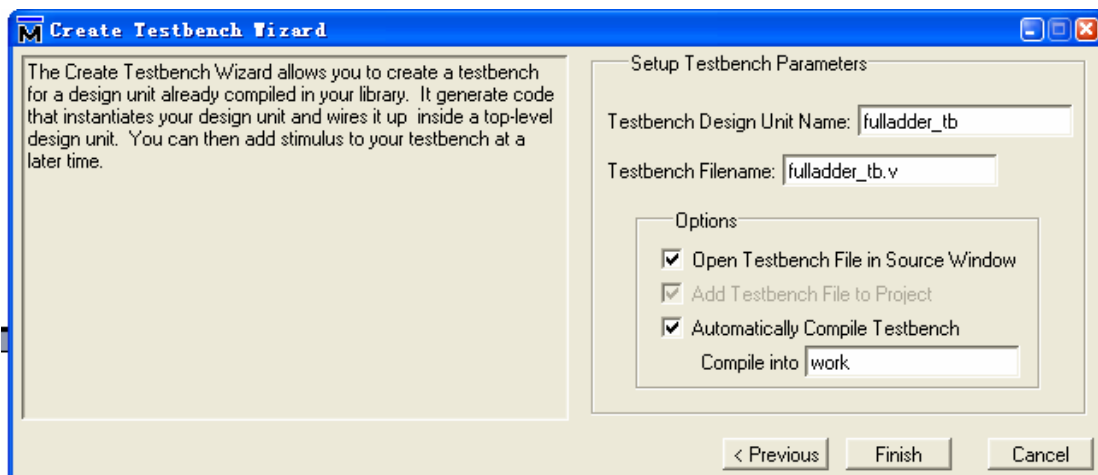
- (1). 执行 File->New->Source->verilog,或者直接点击工具栏上的新建图标,会出现一个 verilog 文档编辑页面,在此文档内设计者即可编辑测试台文件。需要说明的是在 Quartus 中许多不可综合的语句在此处都可以使用,而且 testbench 只是一个激励源产生文件,只要对输入波形进行定义以及显示一些必要信息即可,切记不要编的过于复杂,以免喧宾夺主。
- (2). Modelsim 提供了很多 Testbench 模板,我们直接拿过来用可以减少工作量。点 View->Source->Show Language Templates 然后会出现一个加载工程,接着你会发现刚才的文档编辑窗口左边出现了一个 Language Templates 窗口,见下图。



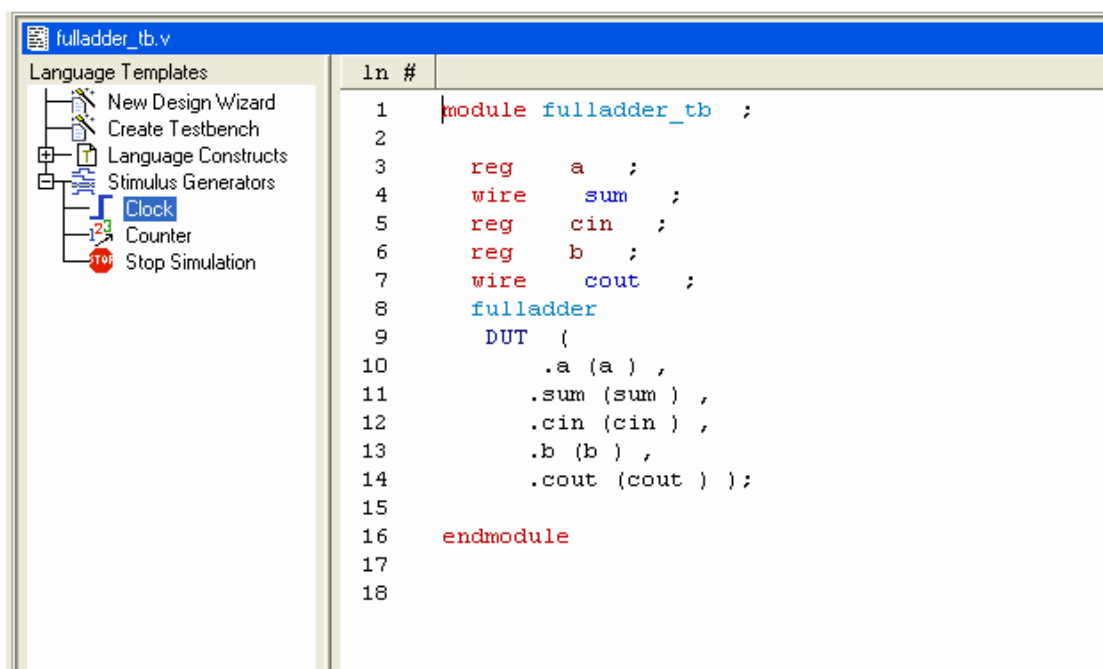
展开 Verilog 项,双击 Create Testbench 会出现一个创建向导,见下图。



选择 Specify Design Unit 工作库下的目标文件，点 Next, 出现下面对话框



可以指定 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 内会出现对目标文件的各个端口的定义还有调用函数




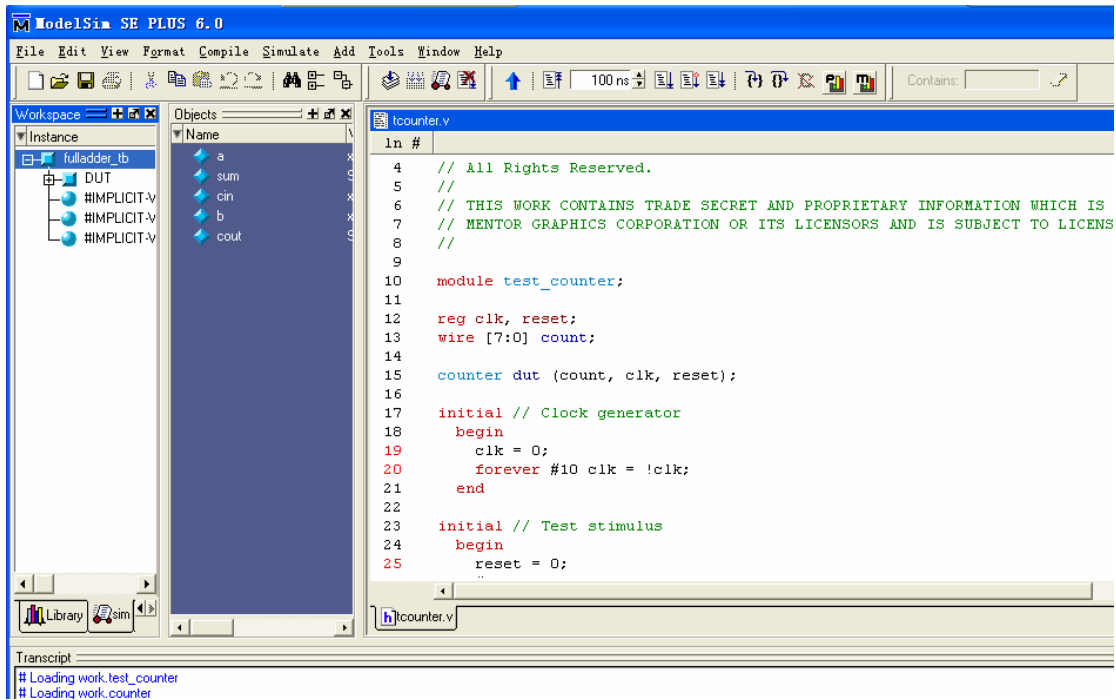
接下来, 设计者可以自己往 Testbench 内添加内容了, 然后保存为 .v 格式即可。按照前面的方法把 Testbench 文件也编译到工作库中。

### 3.3.3 执行仿真

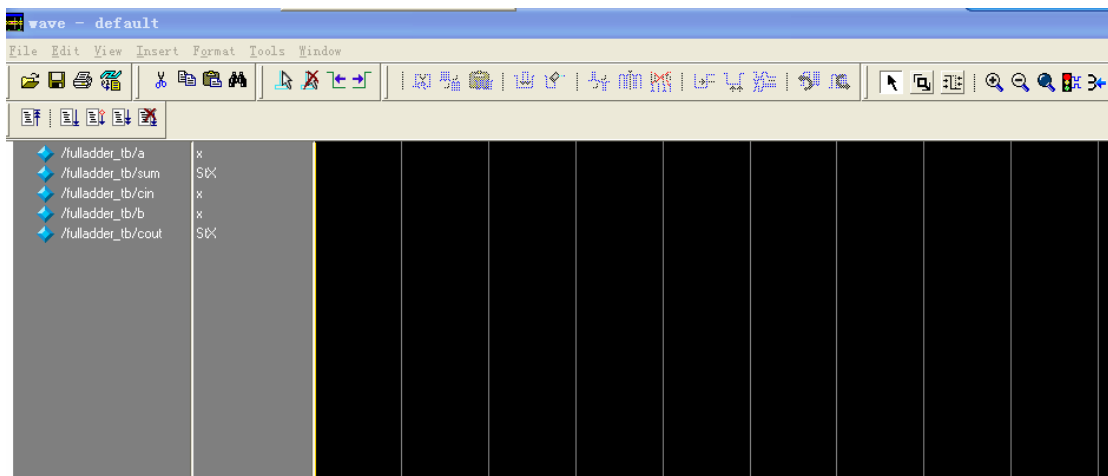
因为仿真分为前仿真和后仿真, 下面分别说明如何操作。


#### (1). 前仿真

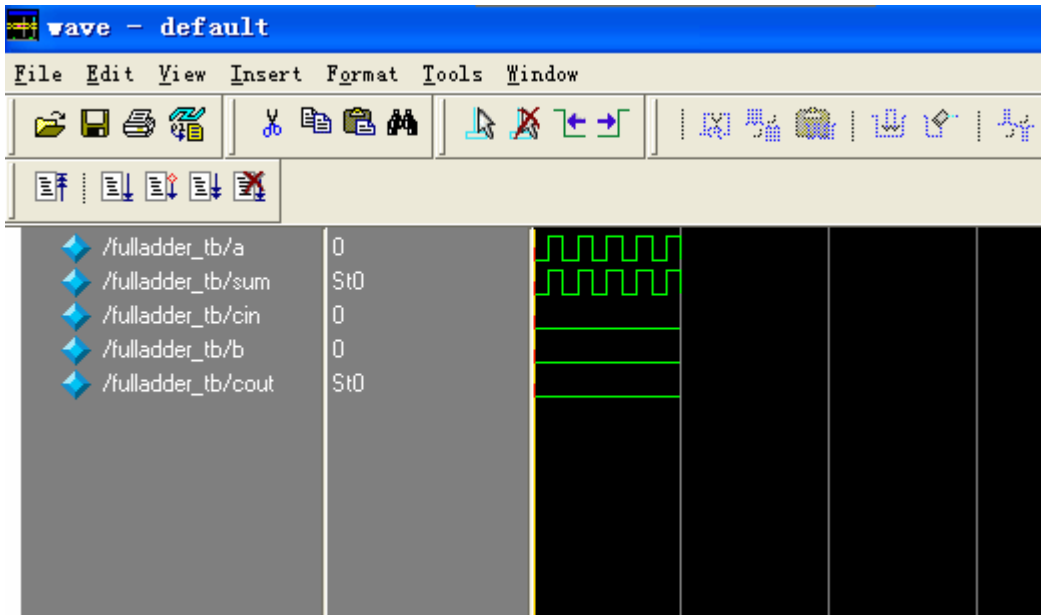
前仿真, 相对来说是比较简单的。在上一步我们已经把需要的文件编译到工作库内了, 现在我们只需点 simulate→Start Simulation 或快捷按钮会出现 start simulate 对话框。点击 Design 标签选择 Work 库下的 Testbench 文件, 然后点 OK 即可, 也可以直接双击 Testbench 文件, 此时会出现下面的界面。



在主界面中会多出来一个 Objects 窗口，里面显示 Testbench 里定义的所有信号引脚，在 Workspace 里也会多出来一个 Sim 标签。右键点击 fulladder\_tb.v, 选择 Add->Add to Wave, 如下图所示。然后将出现 Wave 窗口，现在就可以仿真了，见下图。



窗口里面已经出现了待仿真的各个信号，点击  将开始执行仿真到 100ns, 继续点仿真波形也将继续延伸，见下图。

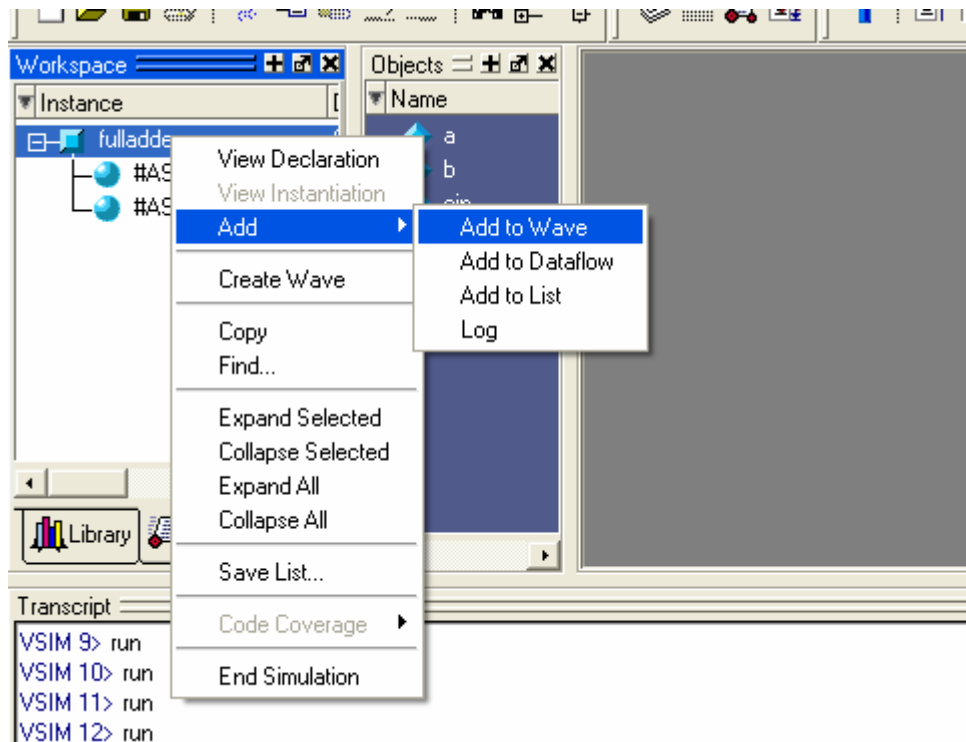


若点, 则仿真一直执行, 直到点才停止仿真。也可以在命令行输入命令:

```
run @1000
```

则执行仿真到 1000ns, 后面的 1000 也可以是别的数值, 设计者可以修改。在下一次运行次命令时将接着当前的波形继续往后仿真。

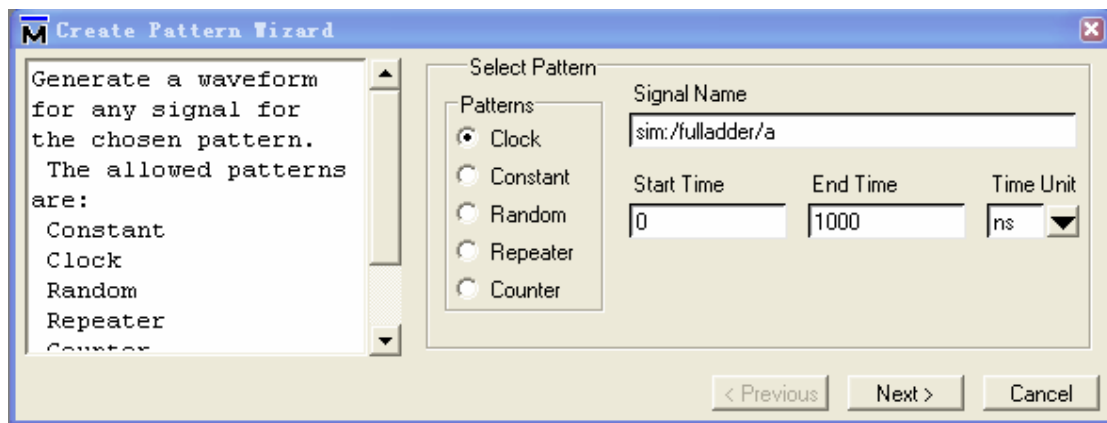
对于复杂的设计文件, 最好是自己编写 testbench 文件, 这样可以精确定义各信号以及各个信号之间的依赖关系等, 提高仿真效率。对于一些简单的设计文件, 也可以在波形窗口自己创建输入波形进行仿真。具体方法是双击 work 库里的目标仿真文件 fulladder.v, 然后点 workspace 窗口中出现的 sim 标签, 右键点击 fuladder, 选择 Add→Add to Wave, 如下图所示。然后将出现 Wave 窗口。



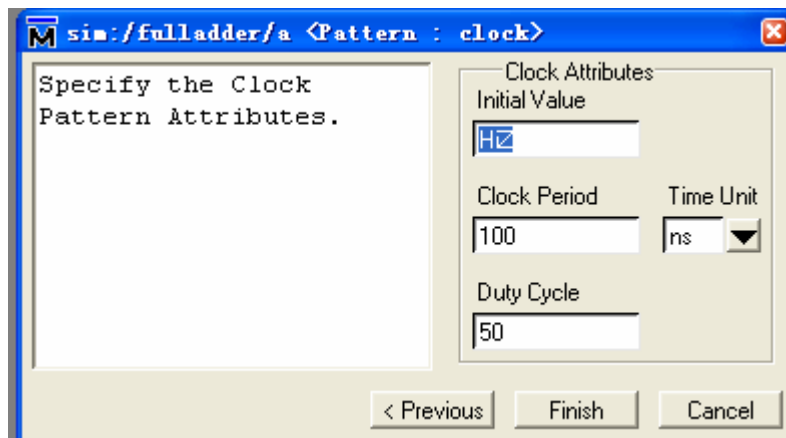
在 wave 窗口中选中要创建波形的信号, 如此例中的 a, 然后右键点击, 选择



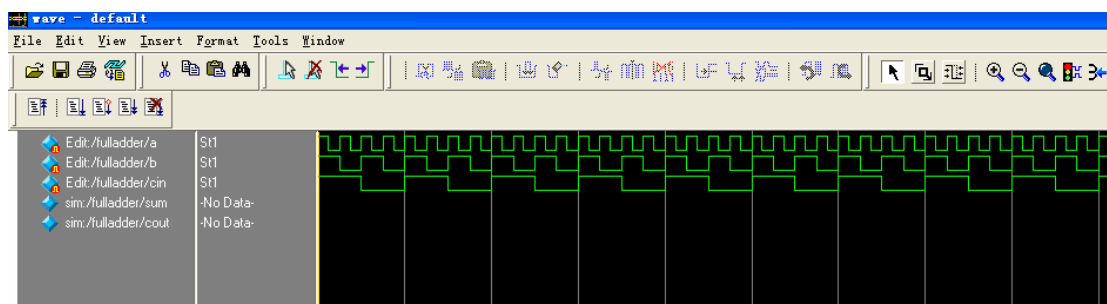
Create/Modify/Wave 项 出现 下 面 的 窗 口 ，



在 Patterns 中选择输入波形的类型，然后分别在右边的窗口中设定起始时间、终止时间以及单位，再点 Next 出现下面的窗口，我们把初始值的 HiZ 改为 0，

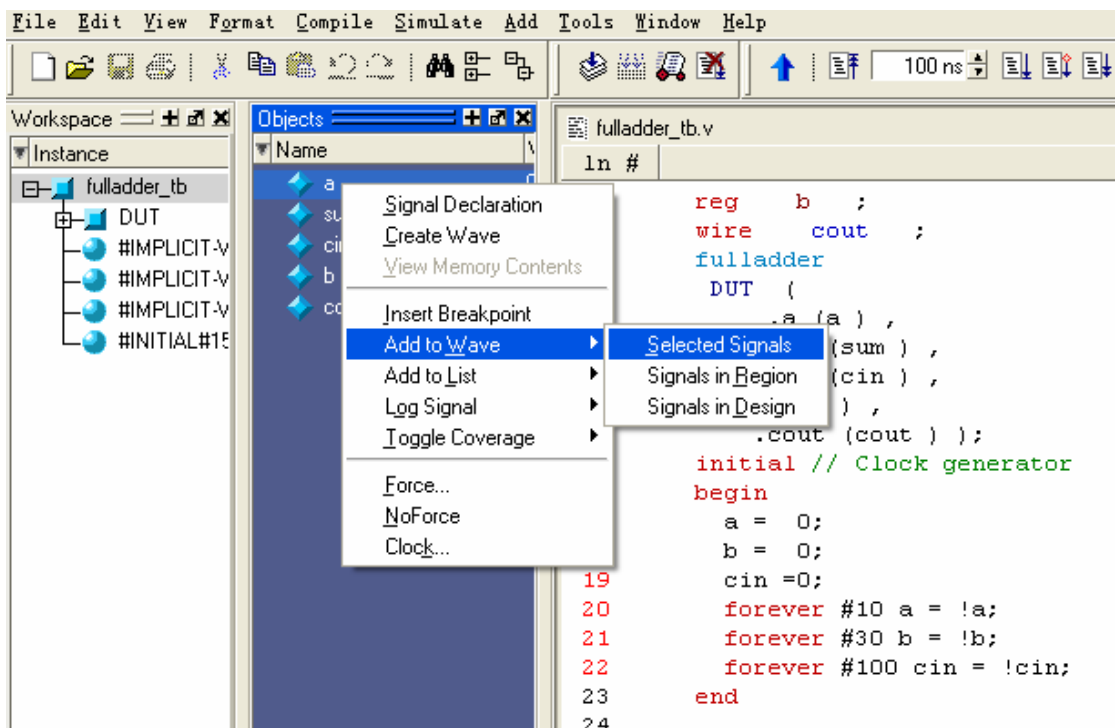


然后修改时钟周期和占空比，然后点 Finish. 接着继续添加其他输入波形，出现下面的结果。前面出现的红点表示该波形是可编辑的。后面的操作与用 testbench 文本仿真的方



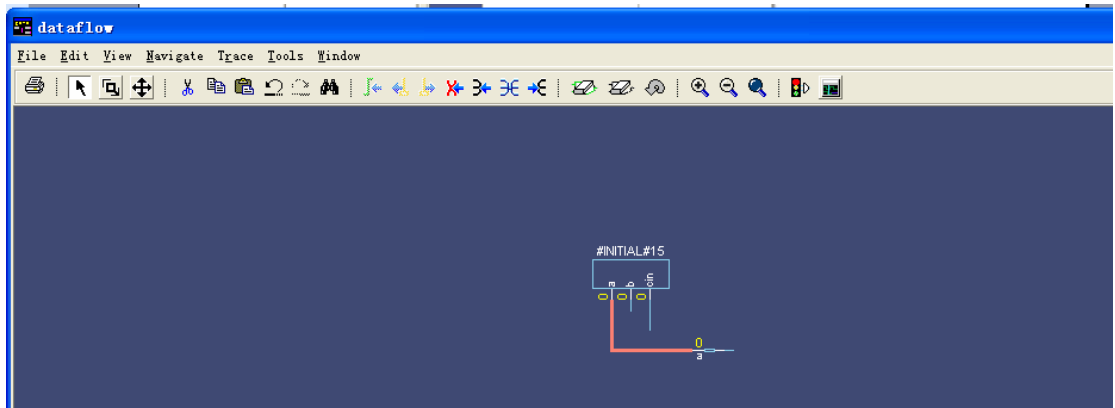
法相同

如果设计者只想查看指定信号的波形，可以先选中 objects 窗口中要观察的信号，然后点右键选择 Add to Wave->Selected signals, 见下图，那么在 Wave 窗口中只添加选中的信号。

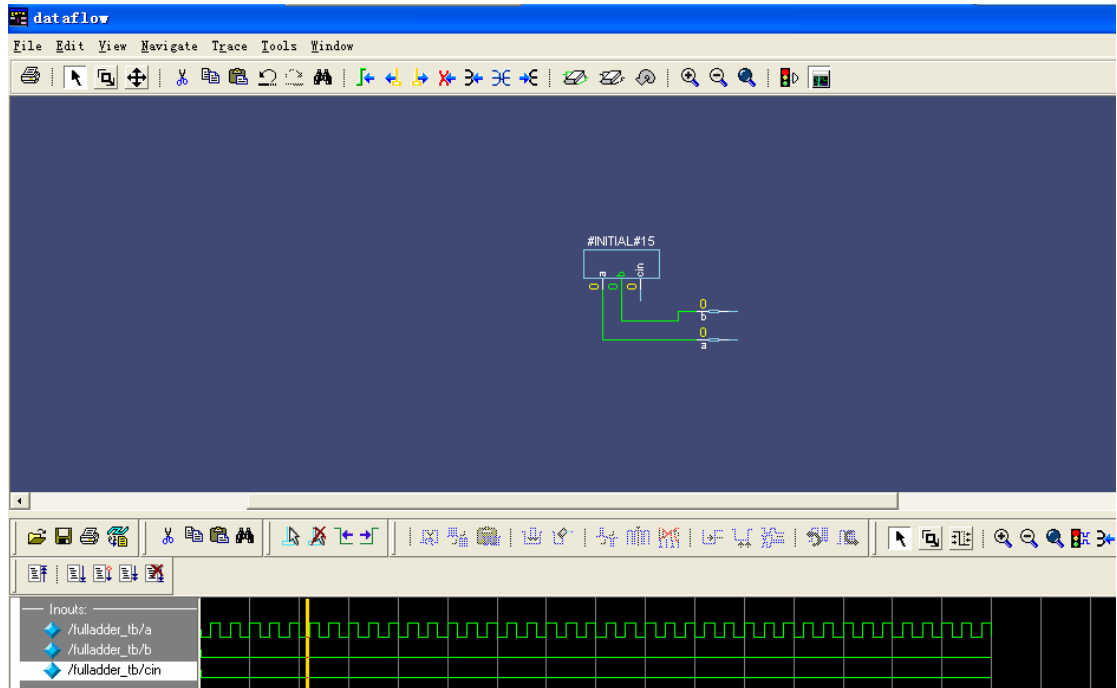


如果要保存波形窗口当前信号的分配，可以点 File→Save→Format, 在出现的对话框中设置保存路径及文件名，保存的格式为.do 文件。如果是想导出自己创建的波形（在文章最后有详细的解释）可以选择 File→Export Waveform 在出现的对话框中选择 EVCD File 并进行相关设置即可，如果导入设计的波形选择 File→Import ECVD 即可。

在主界面中点 View→Debug Windows→Dataflow 可以看到会出现 dataflow 窗口，在 objects 窗口中拖一个信号到该窗口中，你会发现 dataflow 窗口中出现你刚才选中信号所在的模块，如果双击模块的某一引脚，会出现与该引脚相连的别的模块或者引线，见下图。



在 dataflow 窗口中点 View→Show Wave, 会在 dataflow 窗口中出现一个 wave 窗口，双击上面窗口中的某一模块，则在下面的 wave 窗口中出现与该模块相连的所有信号，如果已经执行过仿真，在 wave 窗口中还会出现对应的波形，见下图。



在波形窗口中拖动游标，上面模块的引脚信号的值也会随着游标当前位置的改变而改变。

如果要在 modelsim 中修改原设计文件，在文档页面点击右键，取消 Read Only，即可修改，修改后继续仿真。如果想结束仿真可以点 Simulate->End Simulation, 或直接在命令行输入 quit -sim, 此时 quartus 也会显示结束所有编译过程。

## (2). 后仿真

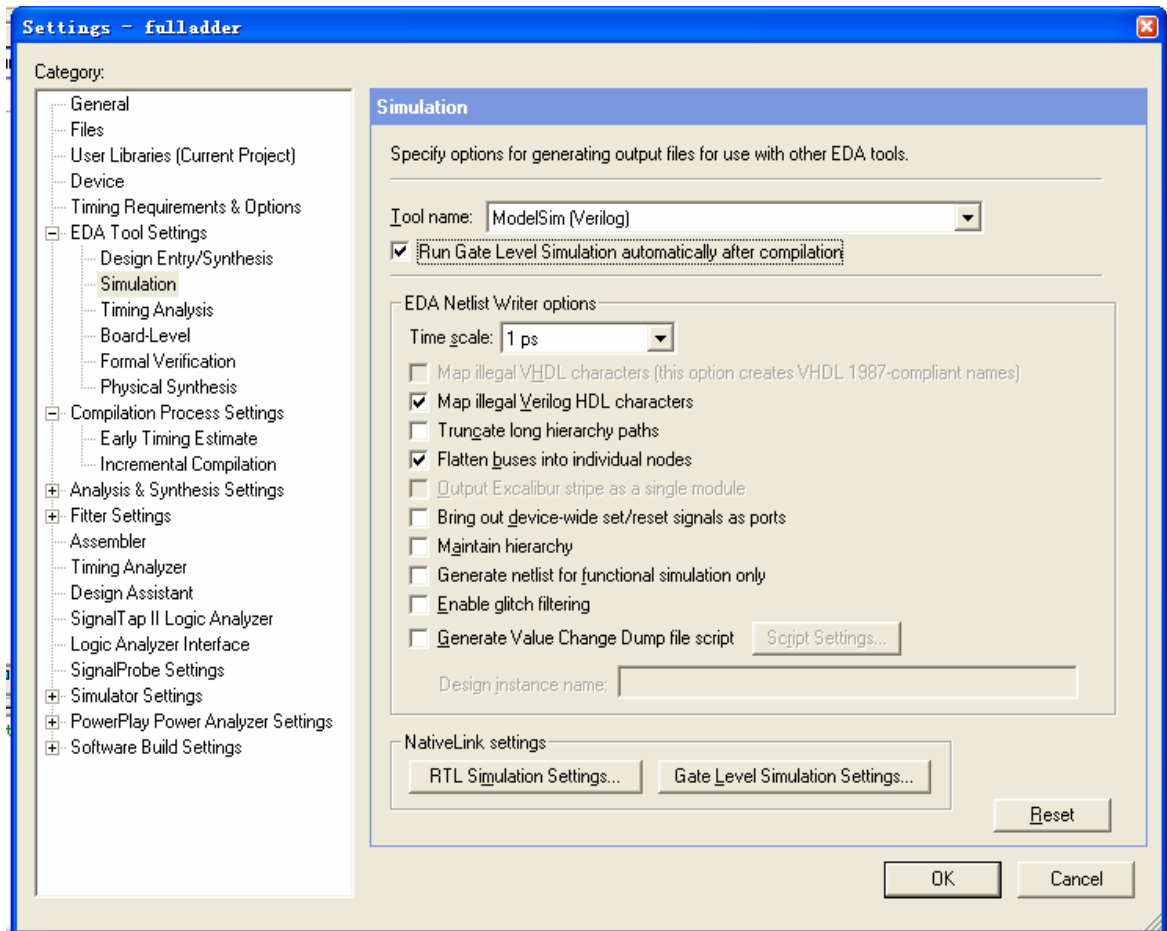
后仿真与前仿真的步骤大体相同，只不过中间需要加添加仿真库、网表和延时文件的步骤。

后仿真的前提是 quartus 已经对要仿真的目标文件进行编译，并生成 modelsim 仿真所需要的 .vo 文件(网表文件)和 .sdo 文件(时延文件)，具体操作过程又有两种方法，一种是通过 Quartus 调用 Modelsim, Quartus 在编译之后自动把仿真需要的 .vo 文件以及需要的仿真库加到 modelsim 中，操作简单；一种是手动将需要的文件和库加入 modelsim 进行仿真，这种方法可以增加主观能动性，充分发挥 modelsim 的强大仿真功能。

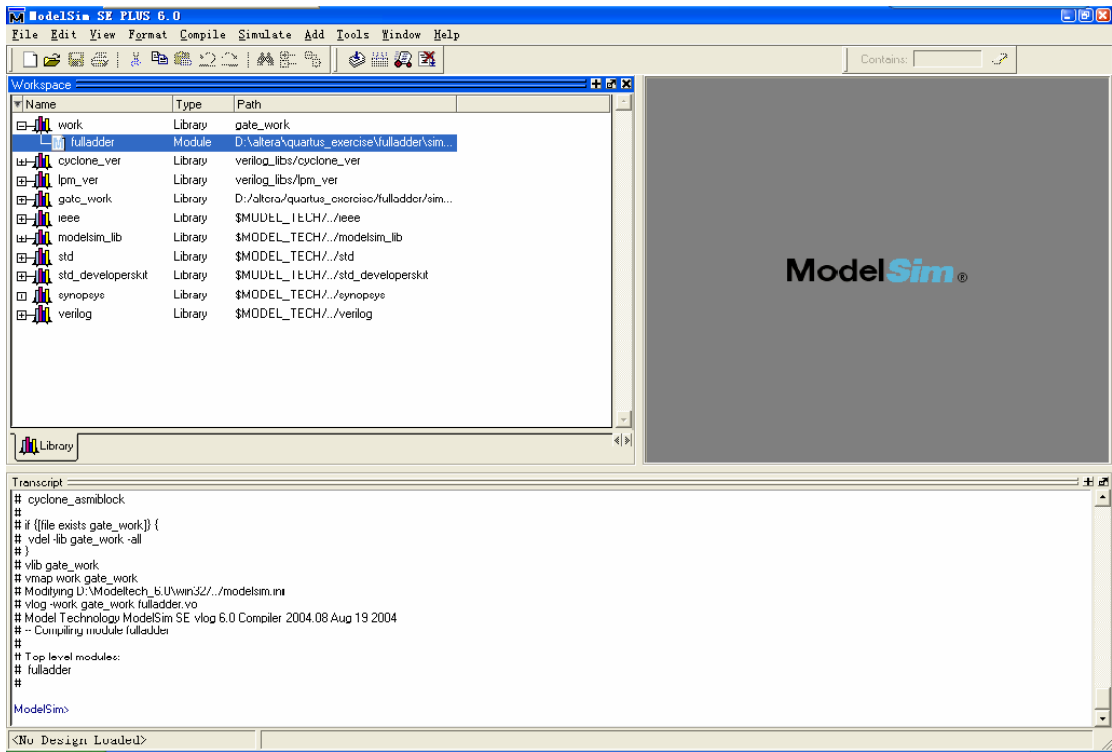
### ① 。通过 Quartus 调用 Modelsim


使用这种方法时首先要对 Quartus 进行设置。

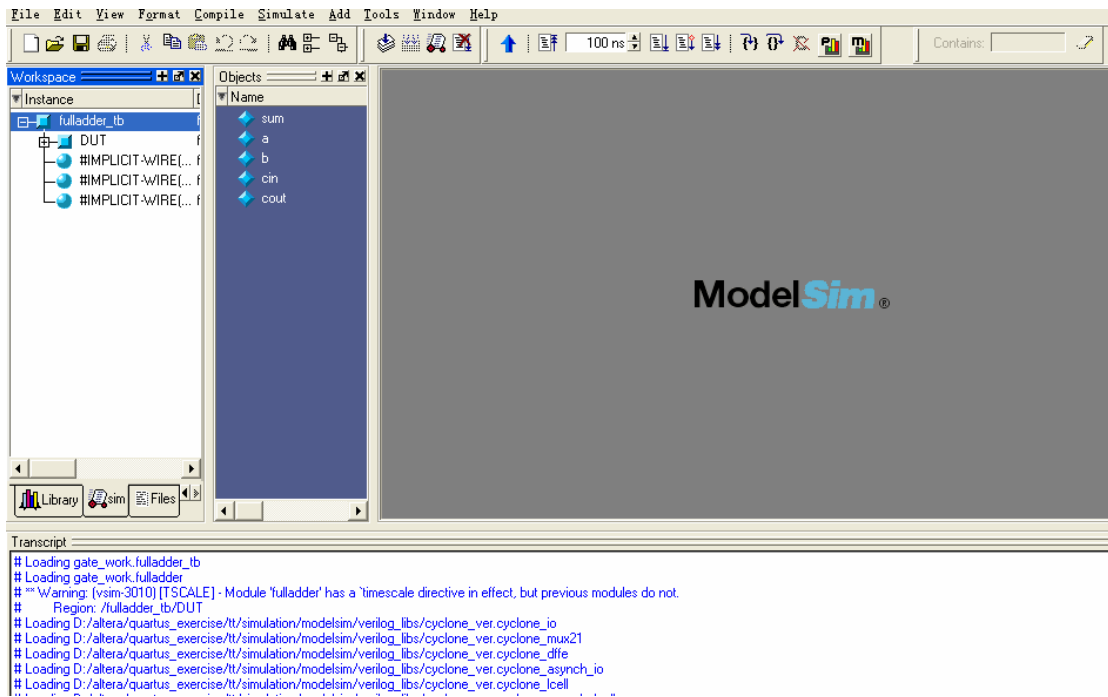
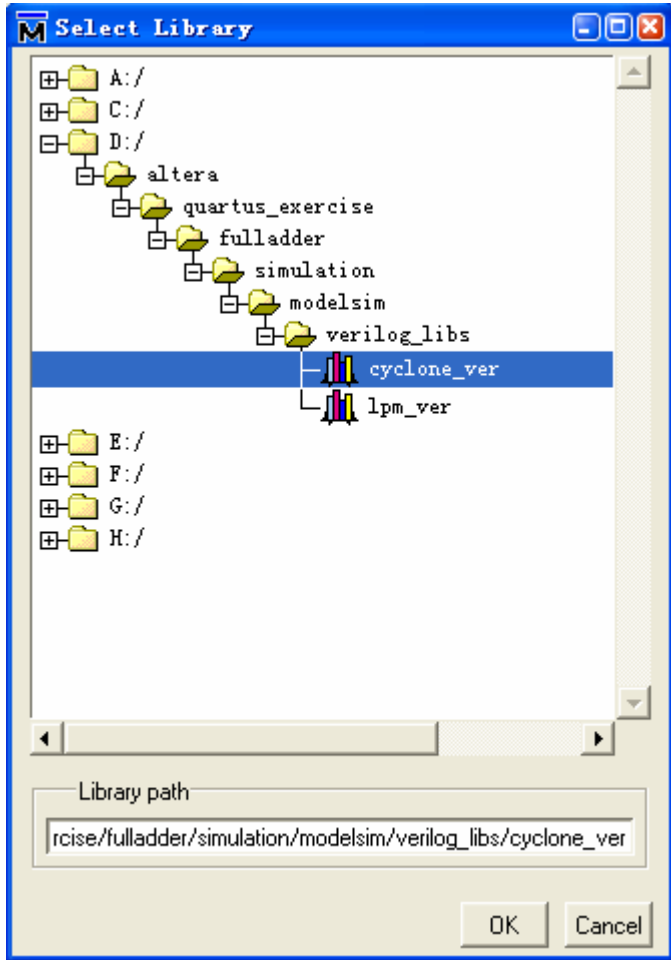
先运行 Quartus, 打开要仿真的工程，点菜单栏的 **Assignments**, 点 **EDA Tool settings**, 选中左边 **Category** 中的 **Simulation.**, 在右边的 **Tool name** 中选 **ModelSim(Verilog)**, 选中下面的 **Run Gate Level Simulation automatically after complication.** 见下图。

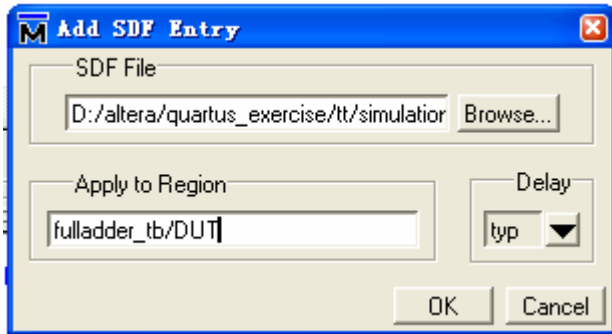


Quartus 中的工程准备好之后点击 **start complication** 按钮，此时 modelsim 会自动启动，而 quartus 处于等待状态（前提是系统环境变量中用户变量中 PATH 要设置好 modelsim 安装路径，如：D:\Modeltech\_6.0\win32）。在打开的 modelsim 的 Workspace 窗口中你会发现多了工作库和资源库，而且 work 库中出现了需要仿真的文件。Modelsim 自动将 quartus 生成的 .vo 文件编译到 work 库，并建立相应的资源库。如图所示。




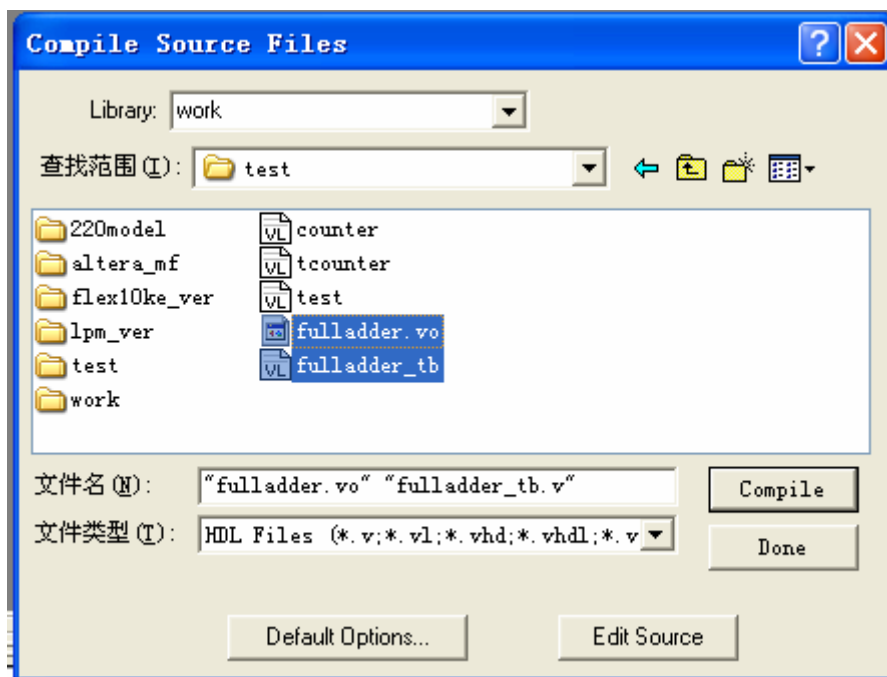
编写测试台程序 `fulladder_tb.v`，最好放在生成的 `.vo` 文件所在的目录，以方便在需要手动仿真时使用。点 **Compile** 在出现的对话框中选中 `fulladder_tb.v` 文件，然后点 **Compile** 按钮，编译结束后点 **Done**，这时在 Work 库中会出现测试台文件。点击 `simulate`→`Start Simulation` 或快捷按钮  会出现 `start simulate` 对话框。点击 **Design** 标签选择 Work 库下的 `fulladder_tb.v` 文件，然后点击 **Libraries** 标签在 `Search Library` 中点击 **Add** 按钮，选择仿真所需要的资源库（如果不知道需要选择哪个库，可以先直接点 **Compile** 看出现的错误提示中说需要的库名，然后再重复上述步骤），见下图。再点 `start simulate` 对话框的 **SDF** 标签。在出现的对话框的 `SDF File` 框内加入 `.sdo` 时延文件路径。在 `Apply To Region` 框内有一个 `/`，在 `/` 的前面输入测试台文件名，即 `fulladder_tb`，在它的后面输入测试台程序中调用被测程序时给被测程序起的名称，本例中为 `DUT`，见下图。然后点 **OK**。后面与前仿真步骤相同。





## ②. 手动仿真

手动仿真需要自己添加文件和编译库，但可以充分发挥 modelsim 强大的仿真功能。操作时也要先对 quartus 进行设置，设置与前面相同只是不要选中 **Run Gate Level Simulation automatically after compilation**。然后启动 modelsim，将当前路径改到新建文件夹下。新建一个库，此处默认库名为 work。如果按照①中的方法编写 testbench 并放在 .vo 所在的目录，这时点 Compile 下的 Compile 或点  将会出现下面的对话框，

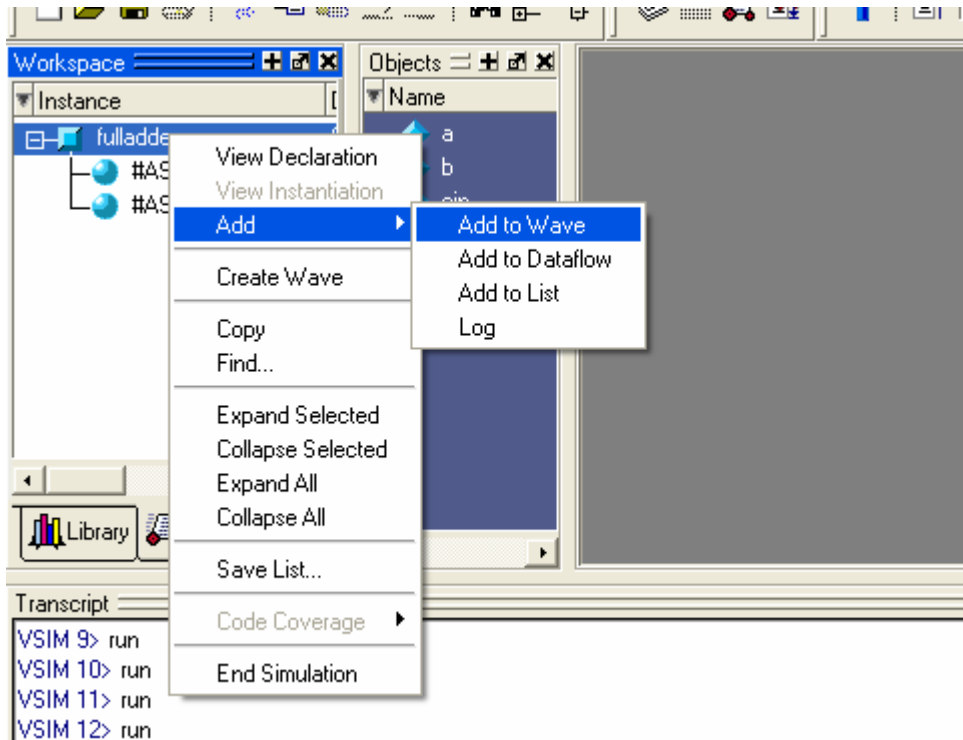


点 Compile 在出现的对话框中同时选中 fulladder\_tb.v 和 fulladder.vo 文件，点击编译，然后点 Done，这样要仿真的网表文件和 testbench 就被编译到工作库里了。后面的操作就与①中的步骤相同了。

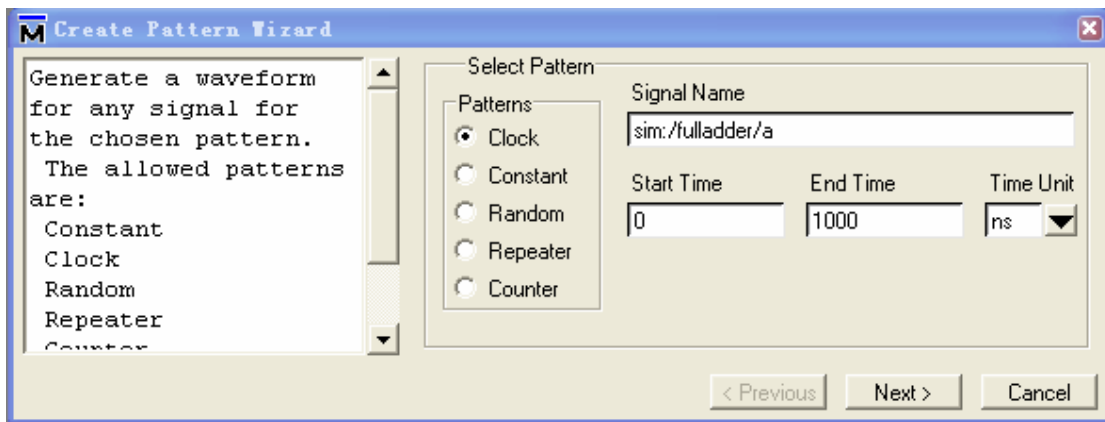
(2). 编写测试台文件，如果只进行功能仿真，双击测试台文件名即可进入仿真界面。如果要进行时序仿真则要将 quartus 生成的 .sdo 文件一起编译到 work 库里，后面的步骤与方法 1 中的相同。

(3). 对于复杂的设计文件，最好是自己编写 testbench 文件，使用行为级监控语法，提高仿真效率。对于一些简单的设计文件，也可以在波形窗口自己创建输入波形进行仿真。具体方法是双击 work 库里的 fulladder.v 文件对其进行仿真，然后点 workspace 窗口中出现

的 sim 标签, 右键电击 fuladder, 选择 Add->Add to Wave, 如下图所示。然后将 出现

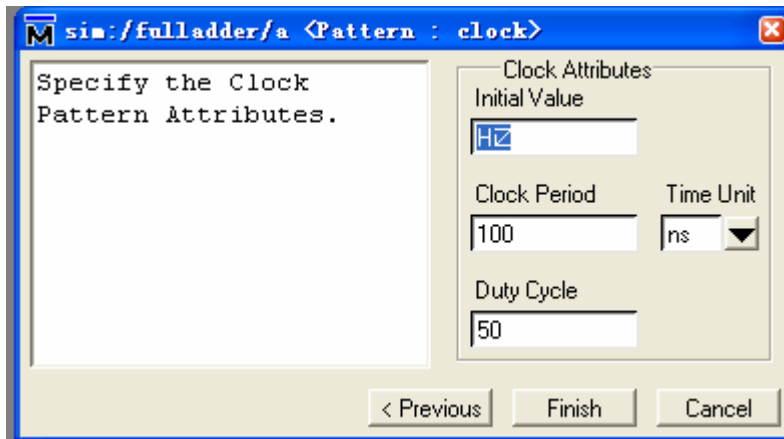


Wave 窗口, 见下图。先选中要创建波形的信号, 如此例中的 a,b,cin, 然后右键点击, 选择 Create/Modify/Wave 项 出现 下面的 窗口 ,

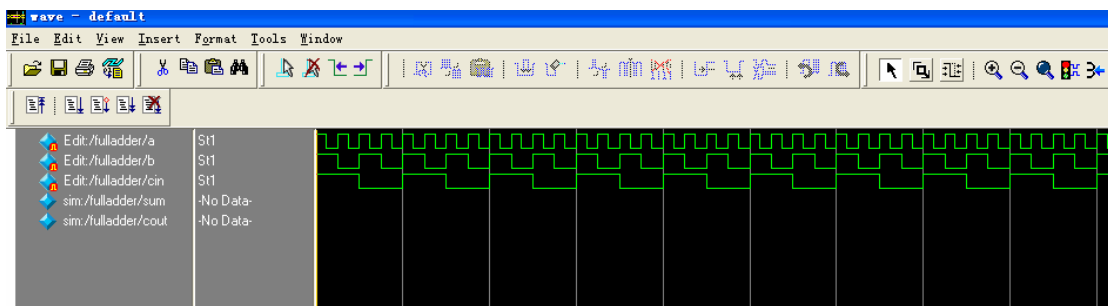



在 Patterns 中选择输入波形的类型, 然后分别在右边的窗口中设定起始时间、终止时间以及单位, 再点 Next 出现下面的窗口, 我们把初始值的

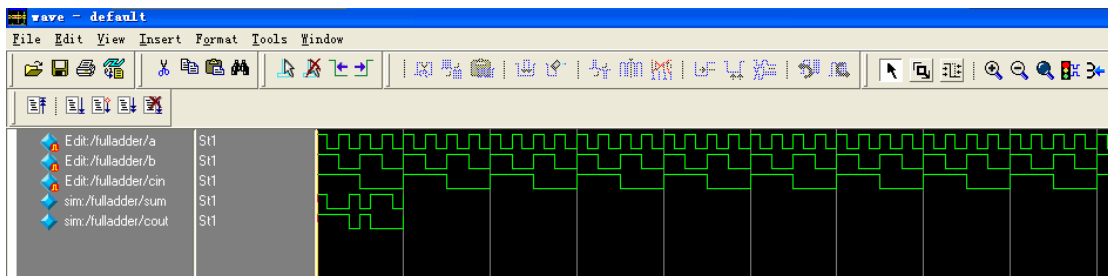





HiZ 改为 0，然后修改时钟周期和占空比，然后点 Finish. 接着继续添加其他输入波形，出现下面的结果。前面出现的红点表示该波形是可编辑的。



然后点主窗口的 Simulate->Run->Run at 100ns 或者直接点  或者在主窗口命令行输入 run @100，则波形将仿真执行到 100ns，如下图。



若点 , 或点 Simulate->Run->Run - all 或者在命令行输入 run @1000, 则仿真到终止时间。

#### 4. 一些说明

##### 4.1 关于在 testbench 里使用 `timescale 的问题

`timescale 是编译器指令，用来定义时延精度和时延单位，命令格式为

```
`timescale time_unit/time_precision
```

其中 time\_unit 定义时延单位，即后面模块中出现的时延数值的单位，time\_precision 定义时延精度。例如

```
`timescale 1ns/100ps
```

表示时延单位为 1ns，时延精度为 100ps。如果后面有语句

```
#5.22 a=1;
```

此时时延值 5.22ns 应该对应为 5.2ns，因为精度为 0.1ns。

如果设计中多个模块带有自身的 `timescale`，编译时模拟器总是定义在所有模块的最小时延精度上，并且所有模块中的时延都自动地换算为到最小试验精度上。

---