

Simulation 用户手册

(Version 1.0)

深圳市紫光同创电子有限公司

版权所有 侵权必究

文档版本修订记录

序号	版本号	发布日期	修订记录	编制/修改人
1	V1.0	2022.07.12	初始版本	李瑞

名词术语解释

英文名称	中文释义
behavior simulation	行为级仿真
post synthesis simulation	综合后仿真
post pnr function simulation	布局布线后功能仿真
post pnr timing simulation	布局布线后时序仿真

目录

1 SIMULATION 总体介绍	5
2 功能描述	6
2.1 编译仿真库	6
2.2 运行仿真	6
3 用户界面	7
3.1 编译库	7
3.2 运行仿真	9
3.3 仿真设置	10
4 操作流程	15
4.1 编译库	15
4.2 启动仿真	17
免责声明	21

图目录

图 3-1 编译库启动界面.....	7
图 3-2 编译库状态界面.....	8
图 3-3 编译库成功状态提示.....	9
图 3-4 右击 top module 运行仿真.....	10
图 3-5 在 Process 菜单下运行仿真.....	10
图 3-6 Simulation_settings 设置.....	11
图 3-7 Compilation 设置.....	12
图 3-8 Elaboration 设置.....	13
图 3-9 Simulation 设置.....	13
图 4-1 启动仿真库界面.....	15
图 4-2 仿真库设置界面.....	16
图 4-3 编译仿真库.....	16
图 4-4 编译完成提示.....	17
图 4-5 编译库输出.....	17
图 4-6 Simulation 目录下 testbench 文件.....	18
图 4-7 Simulation 目录下设置 Top Module.....	18
图 4-8 右击 top module 运行仿真.....	19
图 4-9 在 Process 菜单下启动仿真.....	19
图 4-10 仿真需要运行的 action.....	20
图 4-11 仿真运行的目录.....	20
图 4-12 生成的脚本目录.....	20

1 Simulation 总体介绍

本软件支持的仿真功能主要包括编译仿真库和启动运行仿真两部分。编译仿真库是指调用 ModelSim 或 QuestaSim 等第三方仿真工具编译 GTP 库和 VOP 库, GTP 库应用于前仿(pnr 前的仿真), VOP 库应用于后仿(pnr 后的仿真)。启动运行仿真是指调用 ModelSim 或 QuestaSim 等仿真工具对用户的 design 进行编译和仿真, 分为 run behavior simulation, run post synthesis simulation, run post pnr function simulation, run post pnr timing simulation 等不同阶段的仿真, 其中 pnr 之前的仿真可能要用到 GTP 库, 而 pnr 后的仿真要用到 VOP 库。

2 功能描述

本软件仿真功能可以方便地让用户启动第三方仿真工具如 ModelSim 或 QuestaSim 等进行仿真，验证功能或时序是否正确。一般仿真的流程包括：

- 1) 创建一个 testbench，编译仿真库，对仿真运行进行相应设置；
- 2) 如果是 post synthesis simulation 或者 post pnr simulation 则还需要运行 Synthesis 得到 vm，或 Generate netlist 得到 sim.v 和 sdf；
- 3) 启动仿真。

2.1 编译仿真库

编译仿真库功能主要对仿真模型GTP和VOP，启动第三方工具如ModelSim或QuestaSim进行编译，形成仿真用的库文件。编译后的库名字分别为usim和vsim以及其他复杂模块仿真库（例如hsst），其中usim属于GTP前仿库，而vsim属于VOP后仿库，复杂模块仿真库可以被usim或vsim调用。除了输出编译的库，编译用的ModelSim或QuestaSim脚本也生成出来，运行脚本可以直接生成库而无需打开本软件。

2.2 运行仿真

启动运行仿真包括：

- 1) run behavior simulation，是综合前行为级仿真
- 2) run post synthesis simulation，是综合后gtp仿真
- 3) run post pnr function simulation，是pnr后的功能仿真
- 4) run post pnr timing simulation，是 pnr 后带 sdf 的时序仿真

3 用户界面

3.1 编译库

编译库时用户点击软件中【tools】菜单下的【Compile Simulation Libraries】，会出现如图界面，点击【compile】即开始编译，或者把 command 中的 tcl 命令在 console 中运行也能编译库。

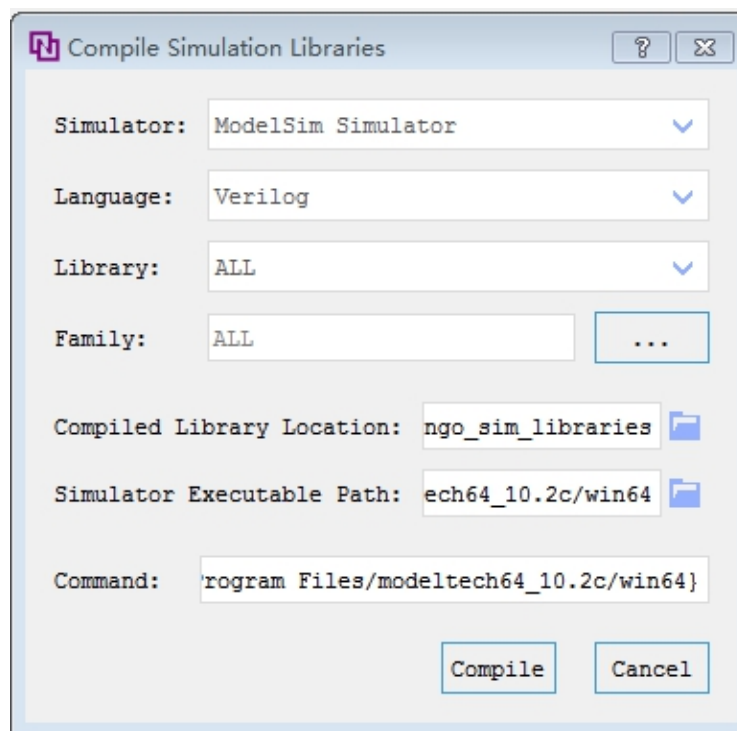


图 3-1 编译库启动界面

各选项含义如下：

- Simulator: 第三方仿真工具，目前支持 ModelSim 和 QuestaSim;
- Language: 仿真库用的语言;
- Library: 选择 usim 则是 GTP 前仿库，vsim 则是 VOP 后仿库，ALL 则包括这两种仿真库;
- Family: 指定芯片系列对应的仿真库进行编译，可支持一次编译多个系列。点击 Family 选项右边的【...】，就会弹出界面用于勾选芯片系列，勾选 ALL 则表示选择所有芯片系列。

- **Compiled Library Location:** 编译出来的库的位置，默认是在当前工程下面；
- **Simulator Executable Path:** 第三方仿真工具运行路径；
- **Command:** 编译库对应的 TCL 命令，可在 Console 中运行。

运行编译库会有状态栏显示正在编译

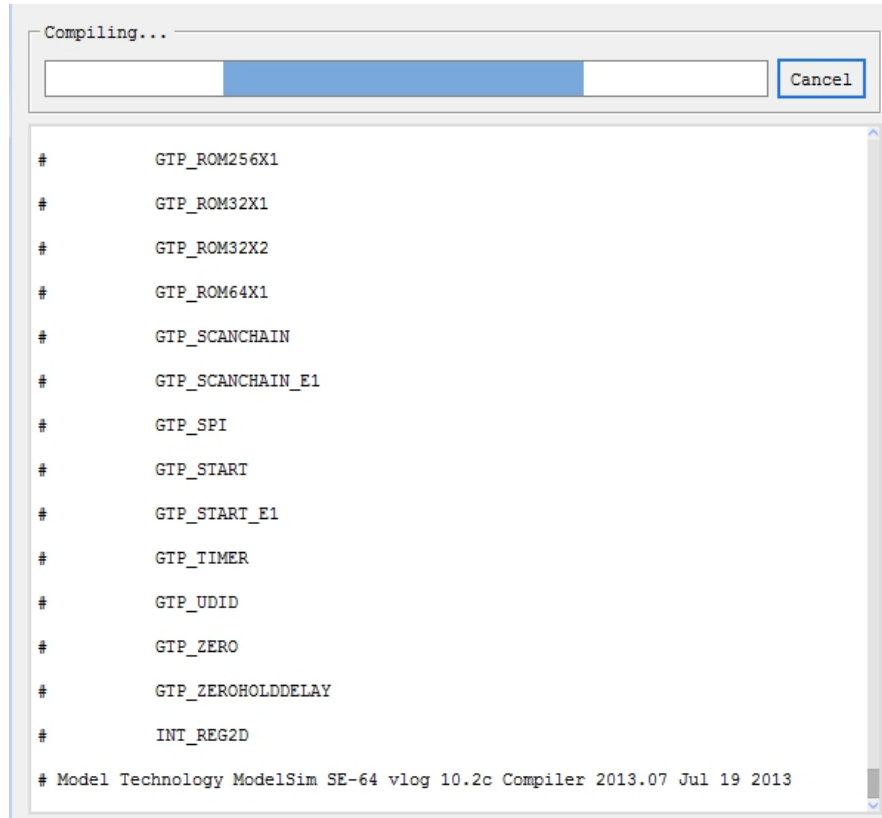


图 3-2 编译库状态界面

编译库完成提示成功如图所示：

```
#
# Top level modules:
# Modifying E:/examples/case1/simulation_library/modelsim.ini
# Model Technology ModelSim SE-64 vlog 10.2c Compiler 2013.07 Jul 19 2013
#
# Top level modules:
# Modifying E:/examples/case1/simulation_library/modelsim.ini
# Model Technology ModelSim SE-64 vlog 10.2c Compiler 2013.07 Jul 19 2013
#
# Top level modules:
# Modifying E:/examples/case1/simulation_library/modelsim.ini
# Model Technology ModelSim SE-64 vlog 10.2c Compiler 2013.07 Jul 19 2013
# ** Warning: ./pciegen3_source_codes/rtl/Adm/radm_outq_mgr.svp(25): in
protected region.
#
# Top level modules:
Compile libraries succeed.
```

图 3-3 编译库成功状态提示

3.2 运行仿真

启动运行仿真是在工程的仿真文件top module下面右击去启动仿真，在【Process】文件菜单下也可以启动。如下图所示是在仿真的top module下面右击出现的菜单选项，包括：

- 1) run behavior simulation，是综合前行为级仿真
- 2) run post synthesis simulation，是综合后gtp仿真
- 3) run post pnr function simulation，是pnr后的功能仿真
- 4) run post pnr timing simulation，是 pnr 后带 sdf 的时序仿真

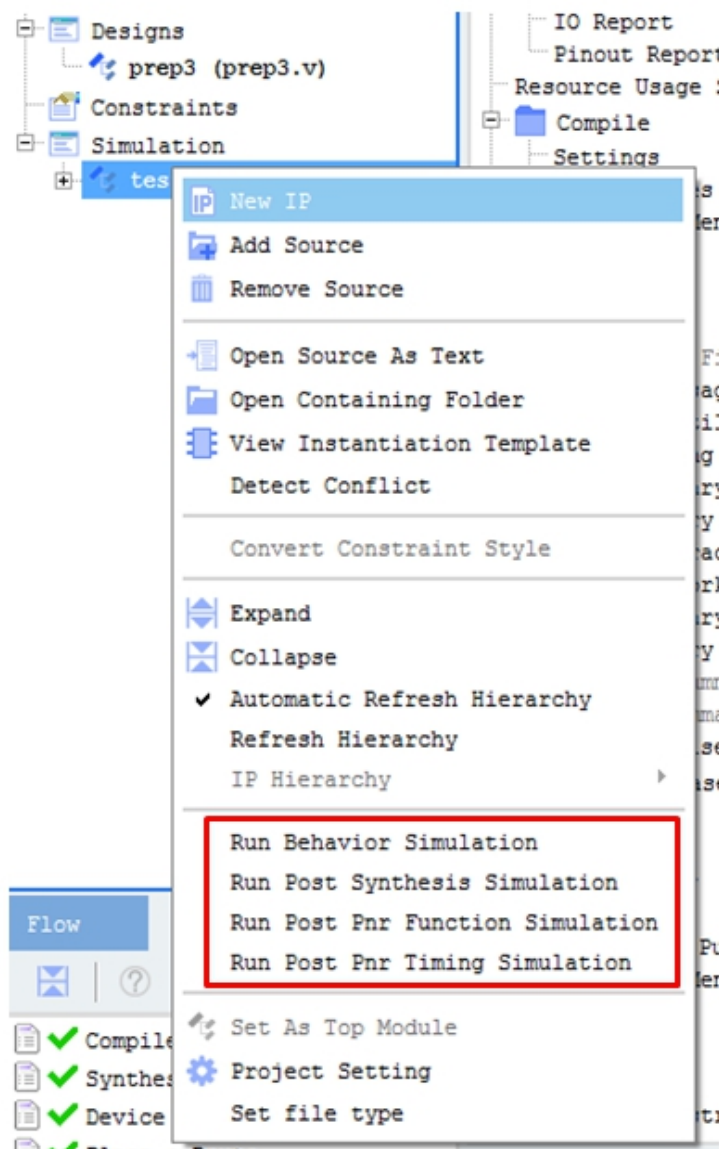


图 3-4 右击 top module 运行仿真

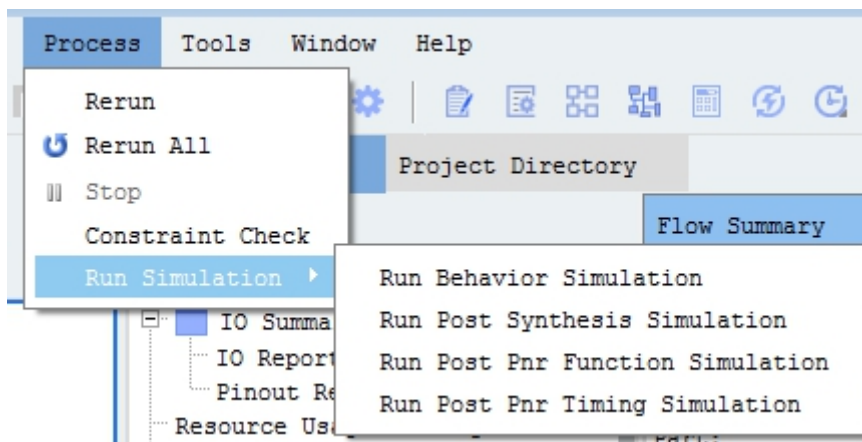


图 3-5 在 Process 菜单下运行仿真

3.3 仿真设置

Simulation_settings 介绍如下：运行仿真有对应的仿真设置，入口为[Project]->[Project

Setting]->[Simulation]。下图所示是目前的仿真设置选项，主要包括仿真工具，top module，library 所在的位置，仿真器路径，以及 Compilation， Elaboration， Simulation 等的设置。

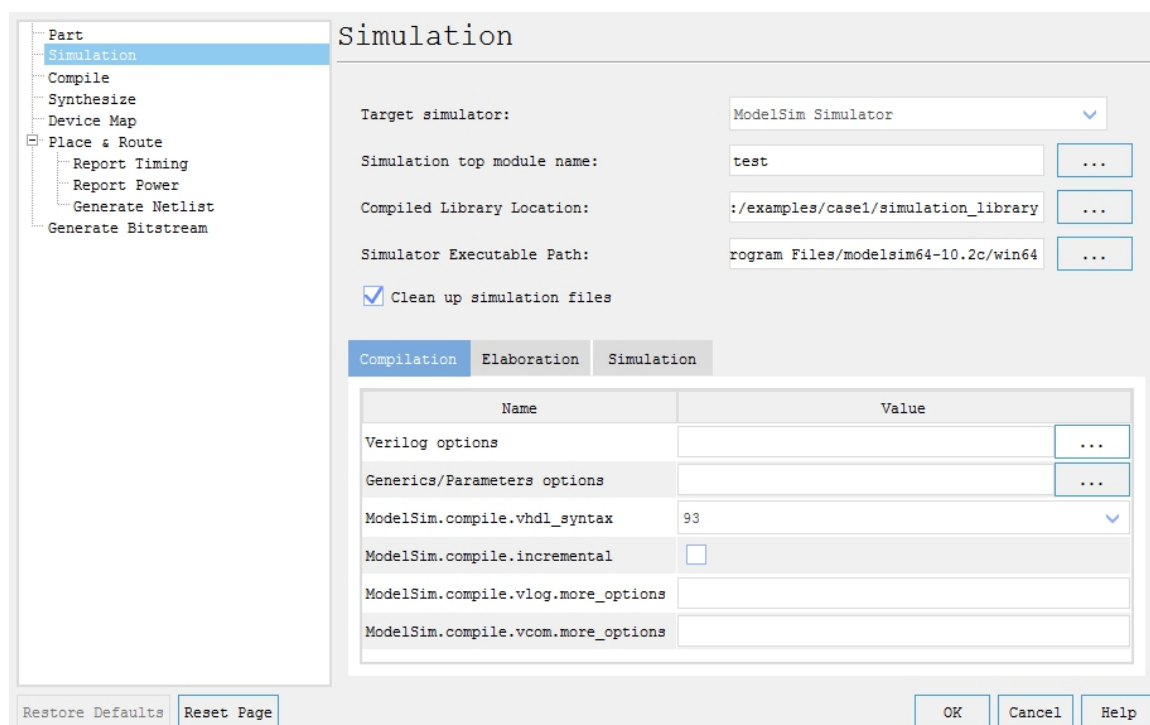


图 3-6 Simulation_settings 设置

各个选项的含义：

- Target Simulator: 目前支持了 ModelSim Simulator 和 QuestaSim Simulator。
- Simulation top module name : 选择仿真的 top module，也可以在仿真文件上右击选择。
- Compile Library Location : 编译好的仿真库的位置，这个会和编译库时选项上的位置保持一致。
- Simulator Executable Path : 设置仿真工具路径。
- Clean up simulation files: 是否运行仿真前先清除仿真文件夹中已有的文件，如果不勾选则会在仿真运行时替换同名文件。

在 Compilation、Elaboration 和 Simulation 三个选项页中，大部分子选项只在名称前缀上有 ModelSim 和 QuestaSim 之分（前缀和 Target Simulator 选项保持一致），其实具体含义都是相同的，所以下面只以 Target Simulator 选项为 ModelSim Simulator 的情况举例：

Compilation 选项下面的子选项含义分别是：

Compilation		Elaboration	Simulation
Name	Value		
Verilog options			...
Generics/Parameters options			...
ModelSim.compile.vhdl_syntax	93		▼
ModelSim.compile.incremental	<input type="checkbox"/>		
ModelSim.compile.vlog.more_options			
ModelSim.compile.vcom.more_options			

图 3-7 Compilation 设置

- Verilog options: 设置 verilog include path 和定义宏。该选项对应 ModelSim 或 QuestaSim 的 vlog 命令的+incdir+选项, +define+选项。
- Generics/Parameters options: 设置 generic/ parameter 的值。对应 vsim 命令的-g 选项。
- Modelsim.compile.vhdl_syntax: 设置支持的 vhdl 语法。对应 vcom 命令的-87 | -93 | -2002 | -2008 选项, 默认为 93。
- Modelsim.compile.incremental: 设置 ModelSim 的编译是否是增量式的。对应 vlog 命令的-incr 选项, 默认不勾选。
- Modelsim.compile.vlog.more_options: 是否增加 vlog 命令更多的选项, 比如-nodebug, -64, 多个选项时需要用空格隔开。
- Modelsim.compile.vcom.more_options: 是否增加 vcom 命令更多的选项, 比如-refresh, -64, 多个选项时需要用空格隔开。

Elaboration 选项下面的子选项含义分别是:

Compilation		Elaboration	Simulation
Name	Value		
Modelsim.elaborate.acc	<input type="checkbox"/>		
Modelsim.elaborate.vopt.more_options	<input type="text"/>		

图 3-8 Elaboration 设置

- Modelsim.elaborate.acc: 使能在做 vopt 优化时能访问到仿真的 Object, 防止被优化。对应 vsim 命令的-voptargs="+acc"选项, 默认不勾选。
- Modelsim.elaborate.vopt.more_options: vopt 命令更多的选项, 比如-64。

Simulation 选项下面的子选项含义分别是:



Compilation	Elaboration	Simulation
Name	Value	
ModelSim.simulate.runtime	<input type="text"/>	
ModelSim.simulate.vcd	test	
ModelSim.simulate_uut	tb2/u_shift_2_steps	
ModelSim.simulate.custom_do	<input type="text"/>	
ModelSim.simulate.custom_udo	<input type="text"/>	
ModelSim.simulate.vsim.more_options	<input type="text"/>	

图 3-9 Simulation 设置

- Modelsim.simulate.runtime: 设置仿真的时间, 对应 ModelSim 的 run xxxns。
- Modesim.simulate.vcd: 生成 vcd 文件的名称, 用于功耗分析
- Modesim.simulate.uut: uut (unit under test) 下面的所有信号的变化会生成到 vcd 文件中。

如图中设置以上两个选项将生成命令

```
vcd file test.vcd
```

```
vcd add -r tb2/u_shift_2_steps/*
```

- **Modelsim.simulate.custom_do:** 使用客户的脚本进行编译，将通过 ModelSim 的 do 命令执行编译，程序不再产生 compile.tcl。
- **Modelsim.simulate.custom_udo:** 使用客户的脚本进行仿真，将通过 ModelSim 的 do 命令在仿真时执行。
- **Modelsim.simulate.vsim.more_options:** 更多 vsim 的选项，比如-sdfmax，sdf 加载 max value。

4 操作流程

4.1 编译库

菜单栏【Tools】按钮点【Compile Simulation Libraries】启动编译库

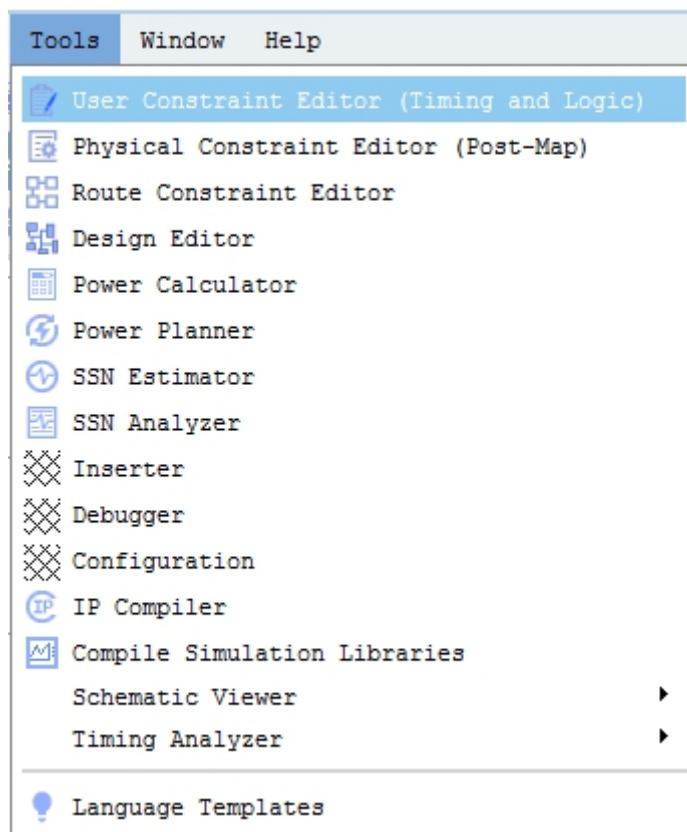


图 4-1 启动仿真库界面

点击后弹出如下配置向导界面：

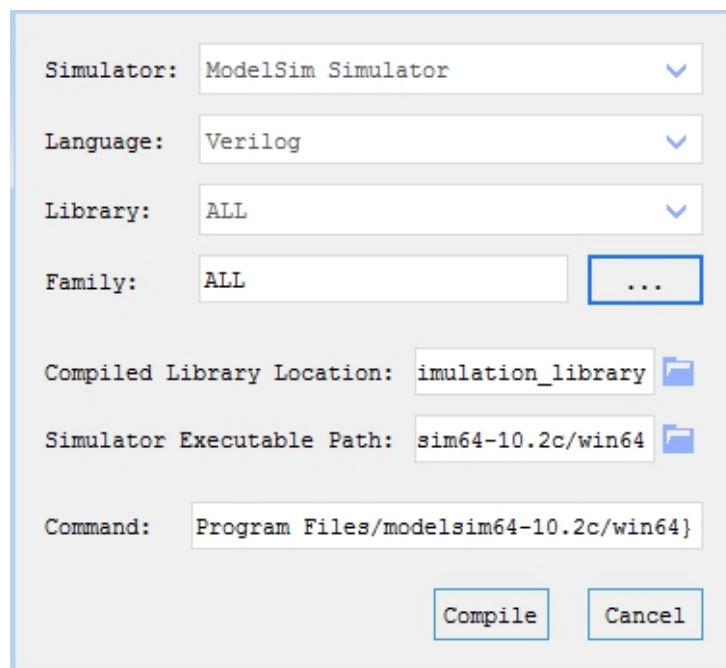


图 4-2 仿真库设置界面

点【Compile】，则出现编译库状态



图 4-3 编译仿真库

点【Cancel】则会取消

当运行完成后则弹出编译成功提示：

```
#
# Top level modules:
# Modifying E:/examples/case1/simulation_library/modelsim.ini
# Model Technology ModelSim SE-64 vlog 10.2c Compiler 2013.07 Jul 19 2013
#
# Top level modules:
# Modifying E:/examples/case1/simulation_library/modelsim.ini
# Model Technology ModelSim SE-64 vlog 10.2c Compiler 2013.07 Jul 19 2013
#
# Top level modules:
# Modifying E:/examples/case1/simulation_library/modelsim.ini
# Model Technology ModelSim SE-64 vlog 10.2c Compiler 2013.07 Jul 19 2013
# ** Warning: ./pciegen3_source_codes/rtl/Adm/radm_outq_mgr.svp(25): in
protected region.
#
# Top level modules:
Compile libraries succeed.
```

图 4-4 编译完成提示

在运行编译库时设置的输出目录下面会生成

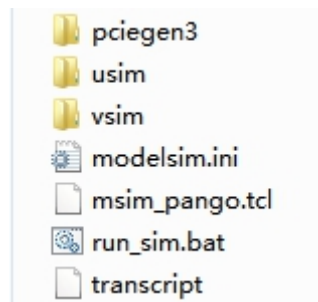


图 4-5 编译库输出

其中 usim, vsim 以及其他复杂模块（例如：pciegen3）是编译生成的库，而 tcl 和 bat 则是编译库的脚本，可以直接双击 bat 生成库而无需运行界面。

4.2 启动仿真

在 Simulation 目录下面添加 testbench 文件，并设置 top module。

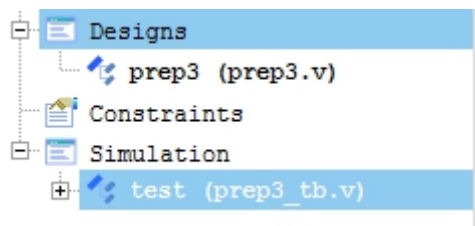


图 4-6 Simulation 目录下 testbench 文件

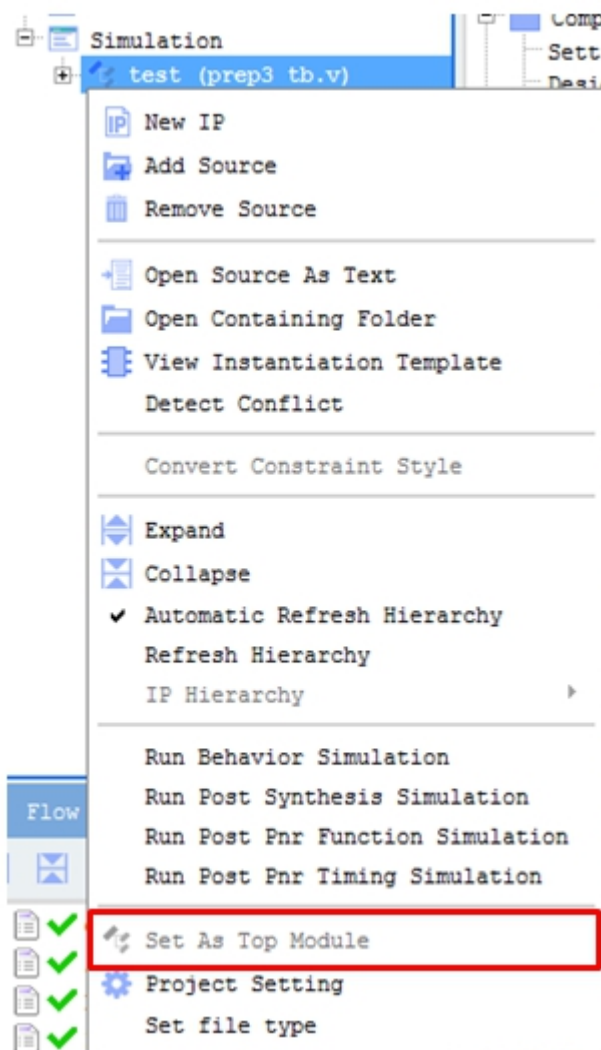


图 4-7 Simulation 目录下设置 Top Module

右击 top module 或者在【Process】文件菜单下选择运行仿真：

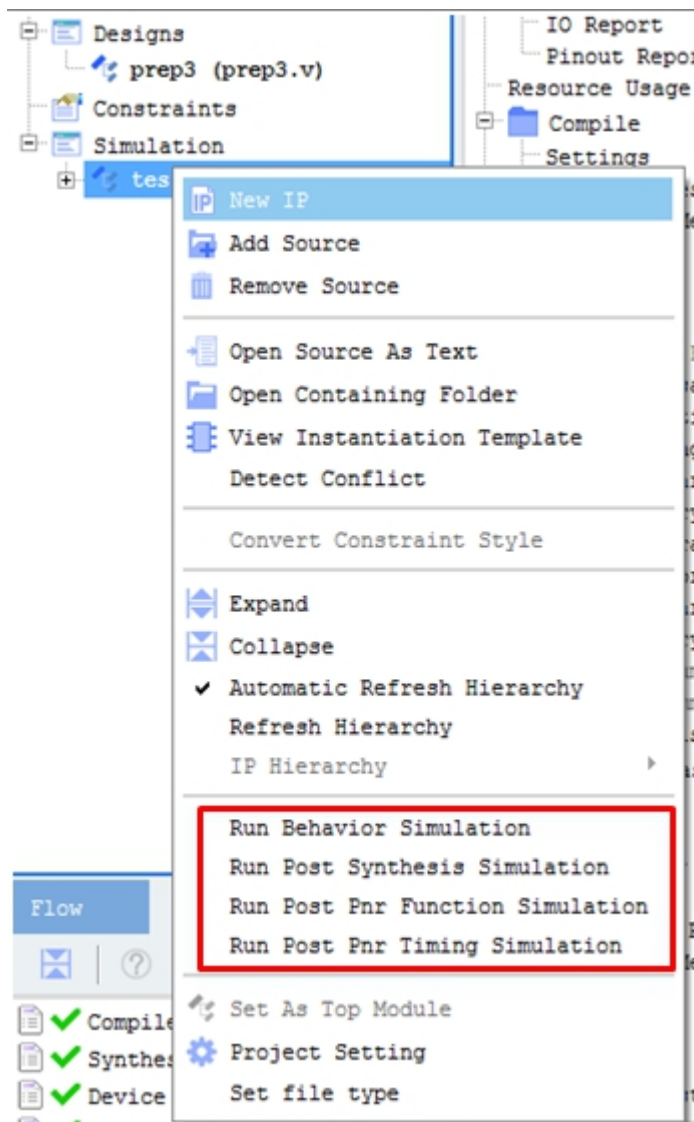


图 4-8 右击 top module 运行仿真

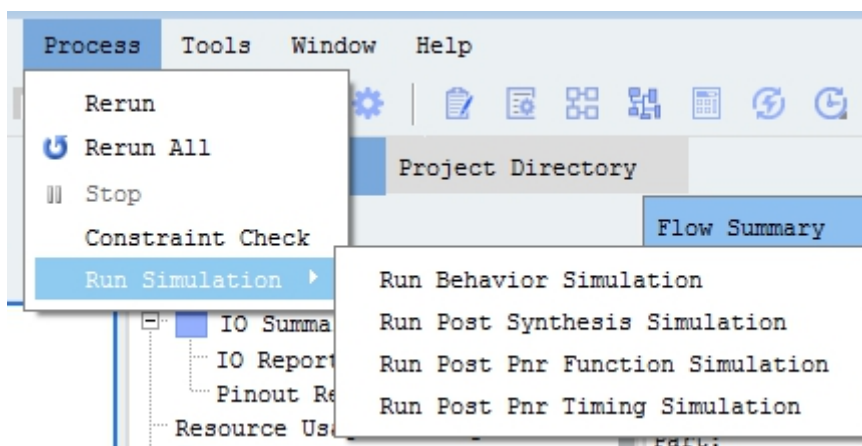


图 4-9 在 Process 菜单下启动仿真

启动 Run Post Synthesis Simulation 需要运行 Synthesis，而启动 Run Post Pnr Function

Simulation 和 Run Post Pnr Timing Simulation 需要生成 sim.v 和 sdf，要运行 Generate Netlist 。

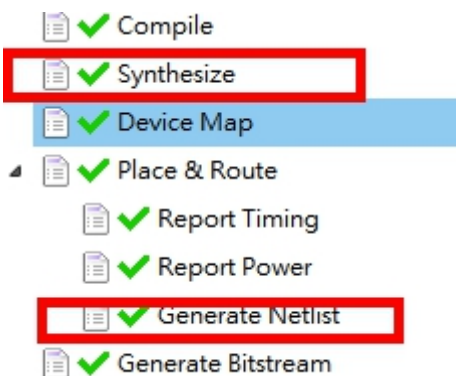


图 4-10 仿真需要运行的 action

不同的仿真设置选项，生成的脚本也会有所不同，生成的脚本在工程目录下面一个叫 sim 的文件夹下面。

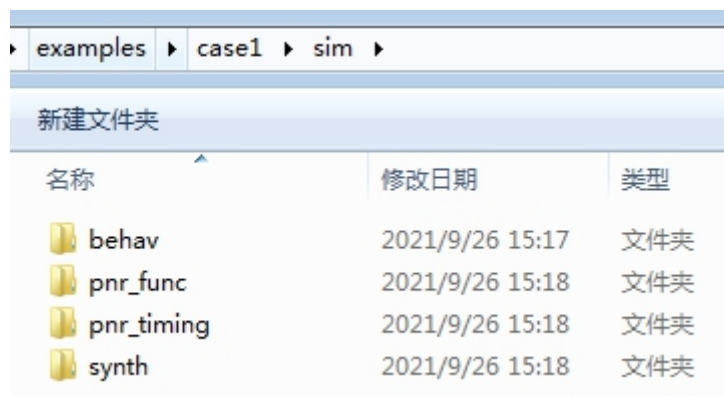


图 4-11 仿真运行的目录

sim 下面 4 个文件夹分别对应 4 个不同阶段的仿真，文件夹里包含仿真脚本等文件。

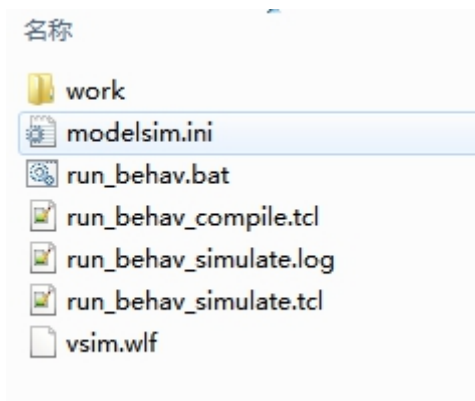


图 4-12 生成的脚本目录

其中*compile.tcl 是编译命令脚本，*simulate.tcl 是仿真运行脚本。双击目录下面的 bat 文件可以直接启动仿真。

免责声明

版权声明

本文档版权归公司所有，并保留一切权利。未经书面许可，任何公司和个人不得将此文档中的任何部分公开、转载或以其他方式披露、散发给第三方。否则，公司必将追究其法律责任。

免责声明

- 1、本文档仅提供阶段性信息，所含内容可根据产品的实际情况随时更新，恕不另行通知。如因本文档使用不当造成的直接或间接损失，本公司不承担任何法律责任。
- 2、本文档按现状提供，不负任何担保责任，包括对适销性、适用于特定用途或非侵权性的任何担保，和任何提案、规格或样品在他处提到的任何担保。本文档在此未以禁止反言或其他方式授予任何知识产权使用许可，不管是明示许可还是暗示许可。
- 3、公司保留任何时候在不事先声明的情况下对公司系列产品相关文档的修改权利。