第 3 章

CHAPTER 3

Xilinx ISE 设计流程

本章将介绍 Xilinx ISE 设计流程，设计内容包括 ISE 设计套件介绍、创建新的设计工程、ISE 开发平台主界面及功能、创建并添加新源文件、添加设计代码、设计综合、设计行为仿真、添加引脚约束文件、设计实现、布局布线后仿真、产生比特流文件、下载比特流文件到 FPGA 芯片、生成和烧写 PROM 文件。

本设计流程是在 Xilinx 大学计划提供的 Nexys3 板卡上实现的。

3.1 ISE 设计套件介绍

Xilinx ISE Design Suite 14 是 Xilinx 公司推出的强大的可编程逻辑器件软件开发平台工具，如图 3.1 所示，ISE Design Suite 软件开发平台主要包括：

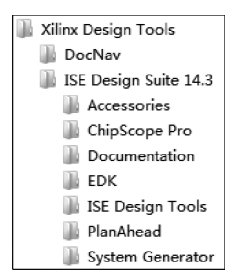


图 3.1 ISE Design Suite 14 设计套件

(1) EDK(嵌入式设计工具)；

(2) PlanAhead(规划设计工具)；

(3) ChipScope Pro(在线逻辑分析仪工具)；

(4) ISE Design Tools(ISE 设计工具)；

(5) System Generator(数字信号处理设计工具)。

下面简单介绍一下这些工具的功能。

(1) ISE 是 Xilinx 公司推出的 FPGA/CPLD 集成开发环境，不仅包括逻辑设计所需的一切，还具有简便易用的内置式工具和向导，使得 I/O 分配、功耗分析、时序驱动设计收敛、HDL 仿真等关键步骤变得容易而直观。

(2) EDK 是 Xilinx 公司推出的 FPGA 嵌入式开发工具，包括嵌入式硬件平台开发工具、嵌入式软件开发工具。可以基于嵌入式 IBM PowerPC 处理器硬核、Xilinx MicroBlaze 处理器软核、ARM Cortex-A9 双核处理器硬核，开发基于 FPGA 的片上嵌入式系统。

(3) System Generator 是 Xilinx 公司推出的简化 FPGA 数字处理系统的集成开发工具，快速、简易地将 DSP 系统的抽象算法转化成可综合的、可靠的硬件系统，为 DSP 设计者扫清了编程的障碍。System Genetator 和 Mathworks 公司的 Simulink 实现无缝连接，在 Simulink 中实现信号的建模、仿真和处理的所有过程。

(4) Chpscope Pro 是 Xilinx 公司推出的在线逻辑分析仪工具，通过软件方式为用户提供稳定和方便的解决方案。该在线逻辑分析仪工具不仅具有逻辑分析仪的功能，而且成本

低廉、操作简单，因此具有极高的实用价值。

Chipscope Pro既可以独立使用，也可以在ISE集成环境中使用，非常灵活，它为用户提供方便和稳定的逻辑分析解决方案，支持Spartan和Virtex全系列FPGA芯片。

Chipscope Pro将逻辑分析器、总线分析器和虚拟I/O小型软件核直接插入到用户的设计中，设计者可以直接查看任何内部信号和节点，包括嵌入式硬核或软核处理器。

(5) PlanAhead工具简化了综合与布局布线之间的设计步骤，能够将大型设计划分成较小的、更易于管理的模块，并集中精力优化各个模块。

此外，还提供了一个直观的环境，为用户设计提供原理图、平面布局规划或器件图，可快速确定和改进设计的层次，以便获得更好的结果和更有效地使用资源，从而获得最佳的性能和更高的利用率，极大地提升了整个设计的性能和质量。

PlanAhead的另一大亮点就是提供了动态可重配置的能力。

Vivado是Xilinx公司推出的新一代集成开发环境。目前，只支持7系列以上FPGA的开发和设计。

思考与练习1　请说明Xilinx ISE设计套件主要包含哪些设计工具？

3.2　创建新的设计工程

本节将介绍创建新的设计工程的步骤，其步骤主要包括：

(1) 打开ISE Project Navigator开发环境主界面，下面给出两种操作方法：①如图3.2所示，在Windows 7操作系统桌面上，鼠标双击Xilinx ISE Design Suite14.3图标。②在Windows 7操作系统的左下角，选择开始→所有程序→Xilinx ISE Design Suite 14.3→ISE Design Tools→Project Navigator。

图3.2　ISE图标

(2) 打开ISE设计软件。在ISE主界面主菜单下，选择File→New Project。

(3) 出现New Project Wizard(新工程向导界面)。对于使用Verilog HDL的设计者，如图3.3所示，按下面参数设置：①Name：gate_Verilog；②Location：D:\EDA_Example\gate_verilog；③Work Directory：D:\EDA_Example\gate_Verilog。对于使用VHDL的设计者，如图3.4所示，按下面参数设置：①Name：gate_VHDL；②Location：D:\EDA_Example\gate_VHDL；③Work Directory：D:\EDA_Example\gate_VHDL；④Top-level source type：HDL。下面对top-level source type下拉框的内容说明如下：①HDL：顶层设计使用HDL语言实现；②Schematic：顶层设计使用原理图实现；③EDIF：顶层设计使用电子设计交换格式(electronic design interchange format，EDIF)实现；④NGC/NGD：顶层设计使用Xilinx的NGC/NGD网表实现。单击Next按钮。

(4) 如图3.5所示，出现New Project Wizard界面，该界面用于选择所使用的FPGA器件的具体型号和设置设计环境参数。按下面参数设置：①product Category(产品范围)：All；②Family(芯片所属系列)：Spartan6；③Device(具体的芯片型号)：XC6SLX16；④Package(封装类型)：CSG324；⑤Speed(速度信息)：－3；⑥Synthesis Tool(综合工具)：XST(VHDL/Verilog)；⑦Simulator(仿真工具)：ISim(VHDL/Verilog)；⑧Preferred Language(设计语言)：Verilog。对于VHDL设计者，选择VHDL。

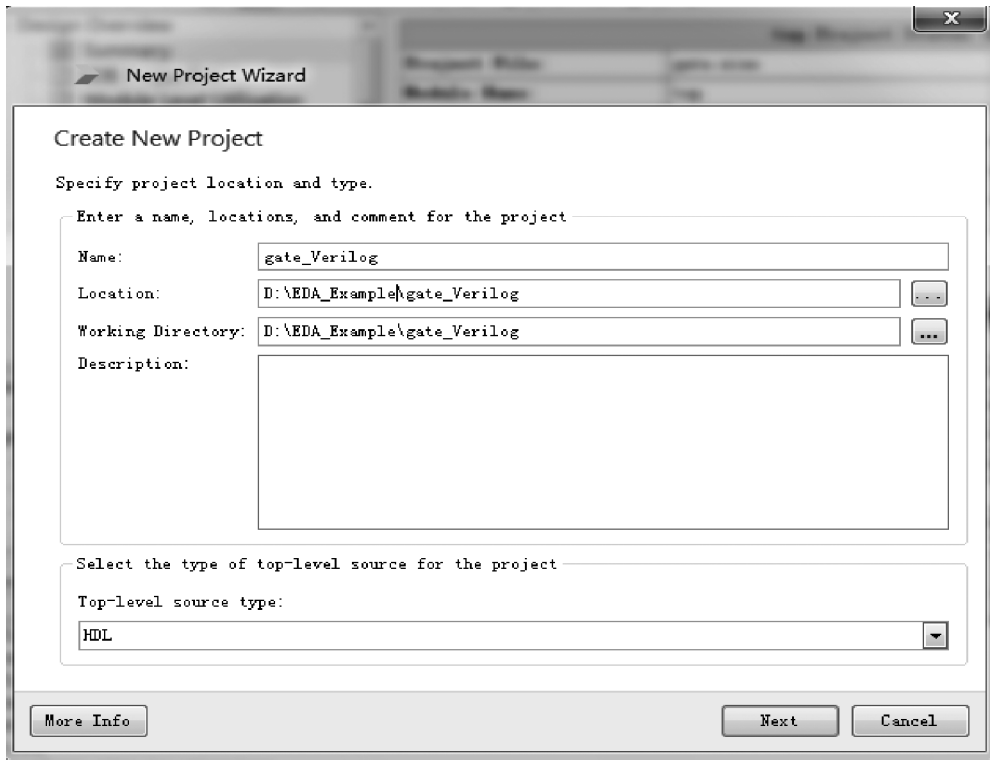


图 3.3　建立工程向导界面(Verilog)

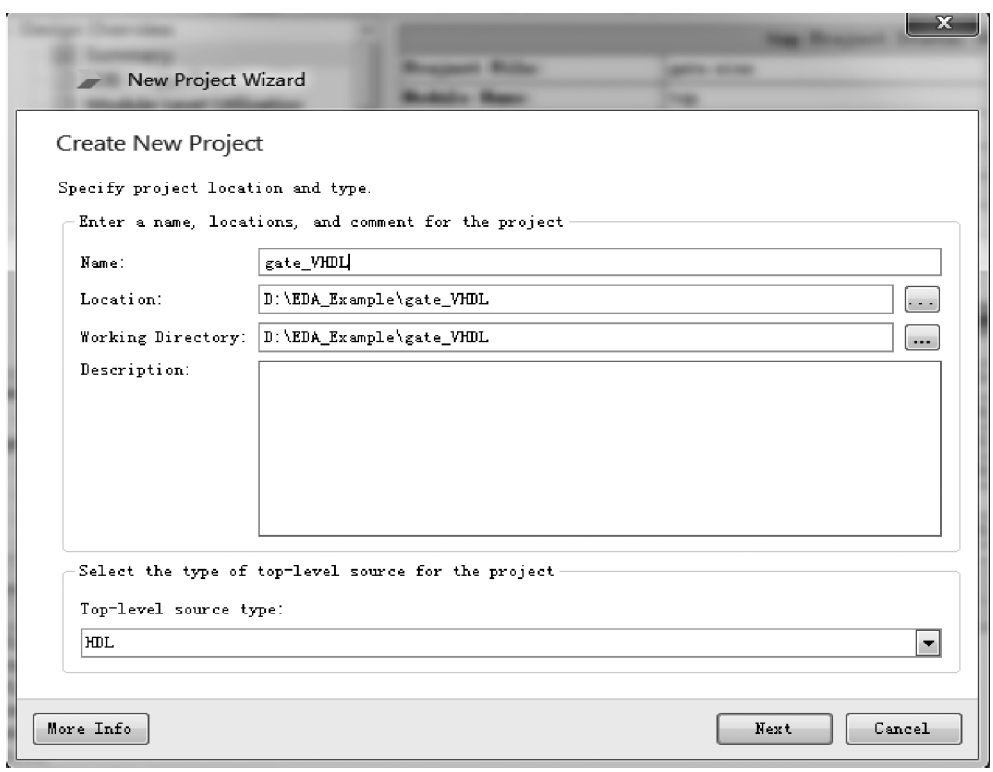


图 3.4　建立工程向导界面(VHDL)

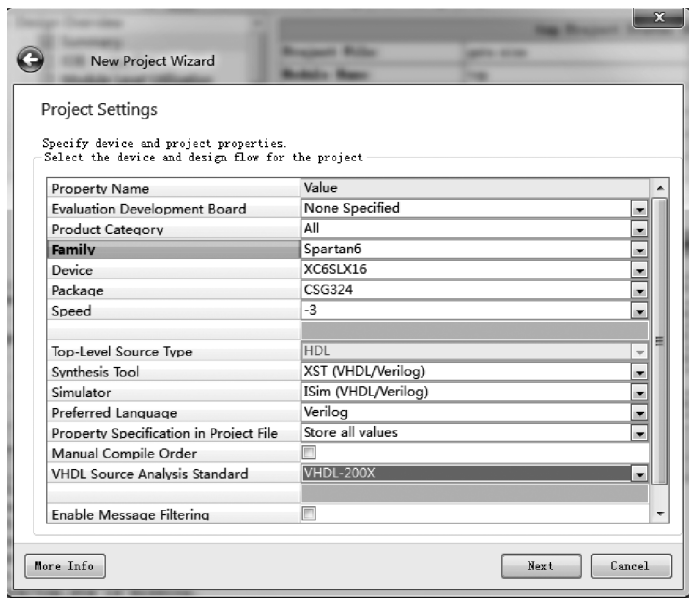


图 3.5　新建工程-设置器件参数和设计参数界面

(5) 单击 Next 按钮。

(6) 出现 New Project Wizard-Project Summary(新工程向导-工程总结)界面,该界面总结了前面设置的参数。

(7) 单击 Finish 按钮。

3.3　ISE 开发平台主界面及功能

ISE 的主界面可以分为 3 个子窗口。

(1) 在主界面窗口的左上面是设计(Design)面板,其中包括:①Start 面板;②Design 面板;③File 面板;④Library 面板。

通过选择不同的面板来显示和访问工程的源文件以及访问当前所选择源文件的运行处理。Start 面板提供了快速访问打开的工程和经常访问的参考资料、文件和教程。

(2) 在 ISE 主界面的底部是控制台(Console)面板,包括:①Console 面板;②Error 面板;③Warnings 面板。显示了状态信息,错误和警告等信息。

(3) ISE 主界面的右边是多文档界面 MDI 窗口,称为工作空间(workspace)。工作空间使设计者可以查看设计报告、文本文件、原理图和仿真波形。每个窗口的大小都可改变,从 ISE 离开;在主界面窗口新的位置,可以平铺、分层或者关闭窗口。

设计者可以在主界面的主菜单下选择 View→Panels 命令,打开或者关闭面板。设计者还可以在主界面的主菜单下选择 Layout→Load Default Layout 恢复默认的窗口布局。

3.3.1　Design(设计)面板

设计面板提供对 View、Hierarchy 和 Processes 面板的访问功能。

1. View 面板

如图 3.6 所示,在 View 面板提供了单选按钮,使设计者能在层次(hierarchy)面板下查看与实现(implementation)或者仿真(simulation)设计流程相关的源文件模块。

如图 3.7 所示,如果设计者选择了 Simulation 单选按钮,则必须从其下方的下拉框中选择一个仿真的阶段:

(1) Behavioral(行为级);

(2) Post-Translate(转换后);

(3) Post-Map(映射后);

(4) Post-Route(布线后)。

图 3.6 设计流程选择

图 3.7 选择仿真的阶段

2. 层次面板

如图 3.8 所示,层次(Hierarchy)面板显示了工程的名字、目标器件、用户文档,以及在图 3.7 中 View 面板内选择设计流程相关的设计源文件。在设计面板中,允许设计者只查看与所选择设计流程(实现或者仿真)相关的那些文件。

在层次面板中的每个文件都有一个相关的图标,图标表示了文件的类型(HDL 文件、原理图、IP 核或者文本文件)。

如图 3.8 所示,如果设计文件包含一个低层次的设计模块,则图标的左边前加“+”符号。通过单击“+”符号,可以展开层次。通过鼠标双击图 3.8 中的文件名字,可以打开文件进行编辑。

3. 处理面板

如图 3.9 所示,处理(Process)面板对上下文敏感。基于在 Source 面板中所选源文件的类型,可以改变处理面板的内容。从处理面板中,设计者可以运行功能,这些功能用来定义、运行和分析设计。处理面板提供了下面的功能:

(1) Design Summary/Report(设计总结/报告)。用于访问设计报告、消息和结果数据的总结,也能执行消息过滤器。

(2) Design Utility(设计工具)。用于访问符号生成、例化模板,察看命令行历史和仿真库编译。

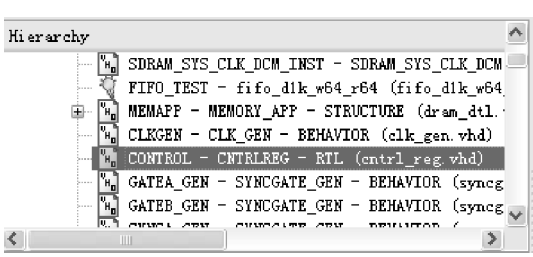


图 3.8 层次面板

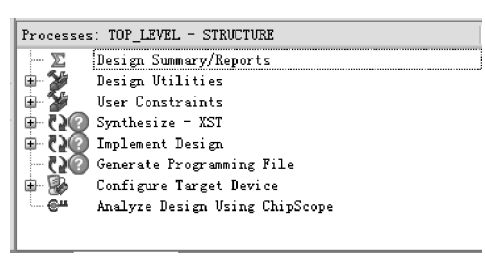


图 3.9 处理面板窗口

(3) User Constraints(用户约束)。用于访问位置和时序约束。

(4) Synthesis(综合)。用于访问检查语法、综合、查看 RTL 原理图和技术原理图,以及查看综合报告,取决于所选择的综合工具,可用的综合过程也不一样。

(5) Implement Design(实现设计)。提供访问实现工具和实现后分析工具。

(6) Generate Programming File(生成编程文件)。访问比特流生成。

(7) Configure Target Device(配置目标器件)。访问配置工具,用于创建可编程的文件和对目标器件编程。

处理面板结合相互依赖的处理技术和工具跟踪所运行的处理过程以及必须要运行的处理。如图 3.9 所示,图形化的状态指示器显示了任意给定时间的流程状态。当选择流程的一个处理时,软件自动运行所需要的处理过程,以便得到所期望的步骤。比如,当设计者运行 Implement Design 处理时,ISE 也运行综合过程,这是由于实现设计过程依赖于最新的综合结果。

查看当前工程命令行参数的运行日志,展开 Design Utility,并且选择 View Command Line Log File。

4. 文件面板

如图 3.10 所示,文件(File)面板提供了一个平面的、排序的工程内所有文件的源文件列表,文件可以通过列表栏的任何一类进行分类。通过使用鼠标单击文件名字,选择 Source Properities 选项,查看每个文件的属性以及修改文件。

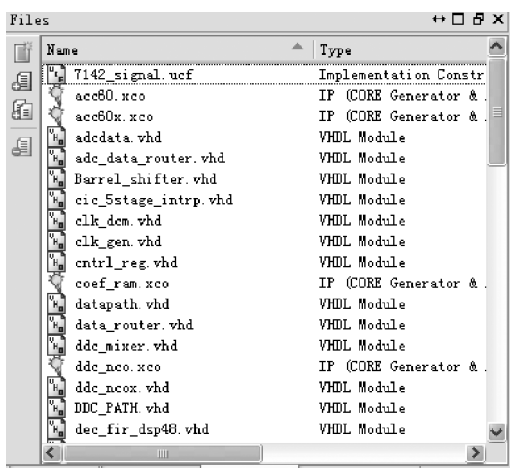


图 3.10　文件面板界面

5. 库面板

库(Library)面板使设计者能管理 HDL 库和它们相关的 HDL 源文件,设计者可以创建、查看和编辑库以及相关的源文件。

3.3.2　Console(控制台)面板

控制台提供了所有来自处理运行的标准输出,窗口显示了错误、警告和消息信息,错误用红色的“x”表示,警告用“!”表示。

1. 错误面板

错误(Error)面板只显示错误信息,过滤掉其他控制台信息。

2. 警告面板

警告(Warings)面板只显示警告信息,过滤掉其他控制台消息。

(1) Error Navigation to Source(错误导航到源代码)。设计者可以从控制台、错误或者警告面板的综合错误或者警告信息导航到 HDL 文件出错的位置。选择错误或者警告信息,单击鼠标右键,出现浮动菜单,选择 Go to Source 选项,就可以打开 HDL 源文件,将光标移动到出错的那一行。

(2) Error Navigation to Record(错误导航到记录)。设计者可以从控制台、错误或者警告面板的综合错误或者警告信息导航到 Xilinx 网站的支持页面相关的回答记录。选择错误或者警告信息,单击鼠标右键,出现浮动菜单,选择 Go to Answer Record 选项,就可以打开默认的 Web 浏览器,显示出对该条信息的所有回答记录。

3.3.3 Workspace

在工作空间(Workspace)可以打开设计编辑器、查看器和分析工具,这些包含 ISE 的文本编辑器、原理图编辑器、约束编辑器、设计总结/报告查看器、RTL 及技术原理图查看器和时序分析器。

其他工具,如用于 I/O 规划和布局的 PlanAhead 软件、Isim 软件、第三方的文本编辑器、XPower 分析器和 iMPACT。当调用这些工具时,可以在 ISE 主窗口界面外独立地打开这些工具。

Design Summary/ReportViewer(设计总结/报告查看器)。设计总结提供了关键设计数据的总结和来自综合和实现工具的所有消息和详细的报告。设计总结列出了关于工程的高级信息,包括信息的概述、器件利用总结、来自布局布线报告所搜集的性能数据、约束信息和来自所有报告连接到各自报告的总结信息,连接到系统设置的报告提供了环境变量的信息和设计实现过程中的工具设置。

3.4 创建并添加新源文件

本节给出创建并添加新源文件的步骤,其步骤主要包括:

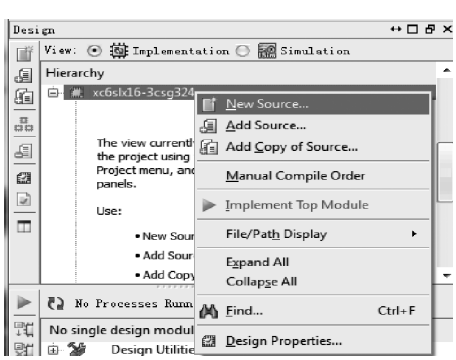


图 3.11 新建一个文件

(1) 如图 3.11 所示,在 Hierarchy 面板内,选中 xc6slx16-3csg324,单击右键,出现浮动菜单,在浮动菜单内选择 New Source。

(2) 如图 3.12 所示,出现 New Source Wizard-Select Source Type(新源文件向导-选择源文件类型)界面。这个界面用于生成新文件,在该界面中选择不同的文件类型,ISE 就可以生成各种源文件的模板。

使用源文件设计向导的好处是设计者不用输入所有的设计代码,只需要修改向导生成的模板文件即可。

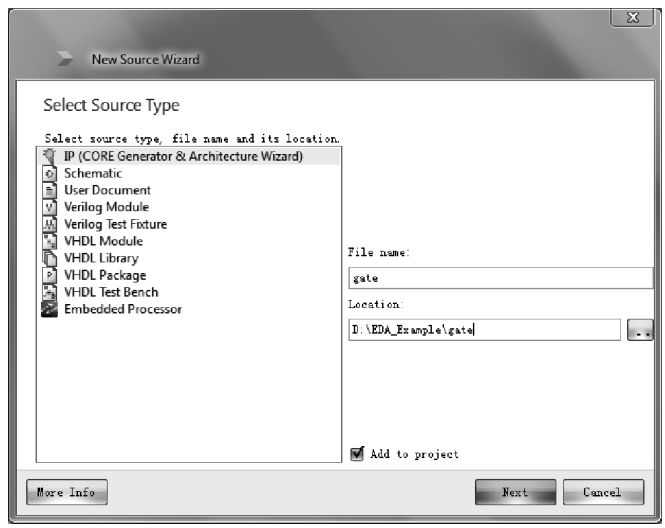


图 3.12　New Source Wizard 界面-选择文件类型

为了方便起见，下面对各个模板的功能进行简要说明。

① IP(CORE Generator & Architecture Wizard)。IP 核生成向导，帮助设计者选择需要的 IP 核类型，并对 IP 核进行配置，然后生成 IP 核。

② Schematic。原理图设计向导，系统的设计采用原理图输入方式实现。

③ System Generator Project。系统生成器工程向导，使用 Xilinx 提供的 System Generator 软件工具和 MATLAB Simulink 工具实现数字信号处理系统的设计。

④ User Document。用户文档，帮助用户编写一些该设计的相关文档说明。

⑤ Verilog Module。Verilog 模块生成向导，帮助设计者生成 Verilog 的 Module 模板。

⑥ Verilog Test Fixture。Verilog 测试平台向导，帮助设计者生成 Verilog 软件仿真测试平台。

⑦ VHDL Module。VDHL 模块生成向导，帮助设计者生成 VHDL 设计模版。

⑧ VHDL Libraray。VHDL 库生成向导，帮助设计者生成 VHDL 语言描述的库模版。

⑨ VHDL Package。VHDL 包生成向导，帮助设计者生成 VHDL 语言描述的包模版。

⑩ VHDL Test Bench。VHDL 测试平台生成向导，帮助设计者生成 VHDL 描述的仿真平台。

⑪ Embedded Processor。嵌入式系统生成向导，帮助设计者在 ISE 工程中插入 EDK 生成的嵌入式设计。

注：设计新文件时，一定要根据设计需要，正确选择生成源文件模板的设计向导。

在该设计中，正确地选择源文件类型：

① 源文件类型：选择 Verilog Module。如果是使用 VHDL 语言完成设计，则在此选择 VHDL Module。

② File(文件名)：gate。

(3) 单击 Next 按钮。

(4) 出现 New Source Wizard-Define Module(新源文件向导-定义模块)界面。

① 对于使用 Verilog HDL 语言的设计者(如图 3.13 所示),按表 3.1 进行参数设置。

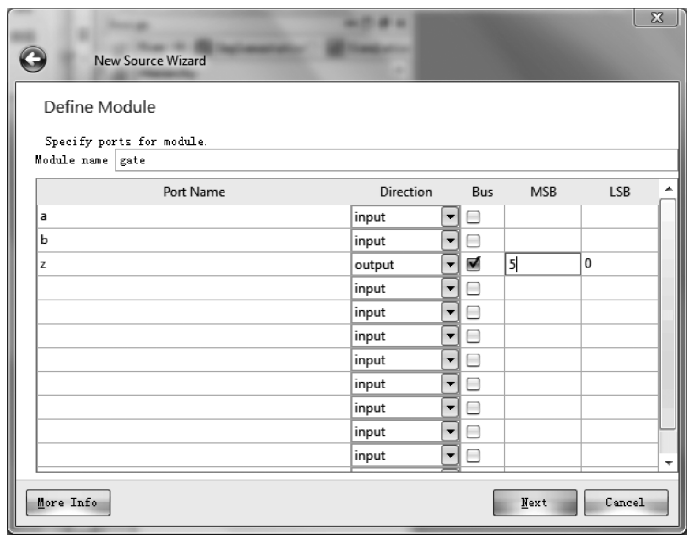


图 3.13 New Source Wizard-Define Module-Verilog 对话框界面

表 3.1 参数端口定义-Verilog

Port Name (端口名)	Direction (方向)	Bus(总线)	
		MSB(最高有效位)	LSB(最低有效位)
a	input	—	—
b	input	—	—
z	output	5	0

② 对于使用 VHDL 语言的设计者(如图 3.14 所示),按表 3.2 进行参数设置。

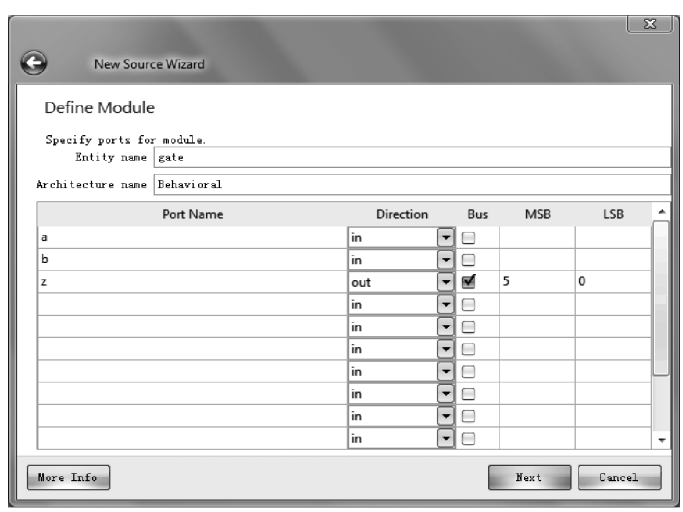


图 3.14 New Source Wizard-DefineModule-VHDL 对话框界面

表 3.2　参数端口定义-VHDL

Port Name（端口名）	Direction（方向）	Bus(总线)	
		MSB(最高有效位)	LSB(最低有效位)
a	in	—	—
b	in	—	—
z	out	5	0

(5) 单击 Next 按钮。

(6) 出现 New Source Wizard-Summary(新源文件向导-总结)界面。

① 对于 VHDL 来说,生成的源文件为 gate. vhd。

② 对于 Verilog HDL 来说,生成的源文件为 gate. v。

(7) 单击 Finish 按钮。

3.5　添加设计代码

本节将在新生成的设计模板中添加设计代码。下面分别介绍 Verilog HDL 设计代码和 VHDL 设计代码的添加。对使用不同语言的读者,分别参考这两部分内容,完成设计代码的添加。

3.5.1　Verilog HDL 设计代码的添加

下面给出添加 Verilog HDL 设计代码的步骤,其步骤主要包括:

(1) 如图 3.15 所示,在 ISE 主界面的左上角的 Hierarchy 面板内,添加 gate. v 文件。此时,ISE 自动打开该文件。

如果没有自动打开该文件,则双击图 3.15 内的 gate(gate. v)图标。

(2) 如图 3.16 所示,ISE 生成了 gate. v 的模板文件,module 为 Verilog HDL 语言的基本设计单元。

图 3.15　gate. v 文件

```
20  //////////////////////////////////////
21  module gate(
22      input a,
23      input b,
24      output [5:0] z
25      );
26
27
28  endmodule
29
```

图 3.16　gate. v 文件模板

(3) 如图 3.17 所示,添加 Verilog HDL 设计代码。

(4) 保存设计代码。

```
20  //////////////////////
21  module gate(
22      input a,
23      input b,
24      output [5:0] z
25      );
26
27   assign z[0]=a & b;
28   assign z[1]=~(a & b);
29   assign z[2]=a | b;
30   assign z[3]=~(a | b);
31   assign z[4]=a ^ b;
32   assign z[5]=a ~^ b;
33
34  endmodule
35
```

图 3.17　添加 Verilog HDL 设计代码

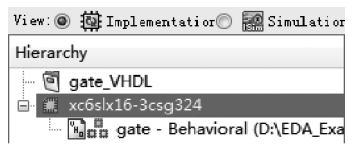


图 3.18　gate.vhd 文件

3.5.2　VHDL 设计代码的添加

下面给出添加 VHDL 设计代码的步骤,其步骤主要包括:

(1) 如图 3.18 所示,在 ISE 主界面的左上角的 Hierarchy 面板内,添加 gate.vhd 文件。此时,ISE 自动打开该文件。

如果没有自动打开该文件,则双击图 3.18 内的 gate(gate.vhd)图标。

(2) 如图 3.19 所示,ISE 生成了 gate.vhd 的模板文件。entity 为 VHDL 语言的基本设计单元。

(3) 如图 3.20 所示,添加 VHDL 设计代码。

(4) 保存设计代码。

```
31
32  entity gate is
33      Port ( a : in  STD_LOGIC;
34             b : in  STD_LOGIC;
35             z : in  STD_LOGIC_VECTOR (5 downto 0));
36  end gate;
37
38  architecture Behavioral of gate is
39
40  begin
41
42
43  end Behavioral;
44
45
```

图 3.19　gate.vhd 文件模板

```
38  architecture Behavioral of gate is
39
40  begin
41    z(0)<=a and b;
42    z(1)<=a nand b;
43    z(2)<=a or b;
44    z(3)<=a nor b;
45    z(4)<=a xor b;
46    z(5)<=a xnor b;
47
48  end Behavioral;
```

图 3.20　添加 VHDL 设计代码

3.6　设计综合

行为级综合可以自动将系统直接从行为级描述综合为寄存器传输级描述。行为级综合的输入为系统的行为级描述,输出为寄存器传输级描述的数据通路。行为级综合工具可以让设计者从更加接近系统概念模型的角度来设计系统;同时,行为级综合工具能让设计者对最终设计电路的面积、性能、功耗以及可测性进行很方便地优化。

3.6.1　Xilinx 综合工具功能

从广义上来说，行为级综合所需要完成的任务包括分配、调度以及绑定。

Xilinx 综合工具在对设计的综合过程中，主要执行以下三个步骤：

(1) 语法检查过程。检查设计文件语法是否有错误。

(2) 编译过程。翻译和优化 HDL 代码，将其转换为综合工具可以识别的元件序列。

(3) 映射过程。将这些可以识别的元件序列转换为可识别的目标技术的术语。

如图 3.21 所示，在 ISE 主界面处理子窗口内的 synthesis 工具可以完成下面的任务：

(1) View RTLschematic (查看 RTL 原理图)；

(2) View Technology Schematic(查看技术原理图)；

(3) Check Syntax (检查语法)；

(4) Generate Post-Synthesis Simulation Model (产生综合后仿真模型)。

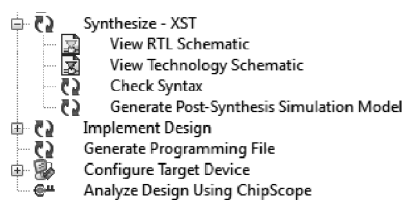


图 3.21　ISE 综合工具

3.6.2　设计综合

本节将介绍设计综合的过程以及查看综合结果的步骤，其步骤主要包括：

(1) 选中要综合的设计文件

① 对于 Verilog HDL 的设计者，在图 3.15 中，选中 gate.v 文件。

② 对于 VHDL 的设计者，在图 3.18 中，选中 gate.vhd 文件。

(2) 在图 3.21 内，双击 Synthesize-XST 选项。此时，XST 工具开始对设计进行综合。

(3) 如图 3.22 所示，双击 Design Summary/Reports，查看报告，了解资源的使用情况。

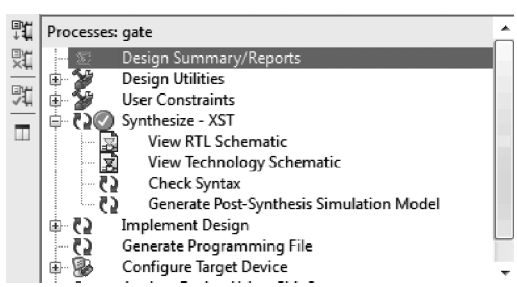


图 3.22　查看综合报告

(4) 查看综合后的门级结构。如图 3.23 所示，双击 View Technology Schematic(查看技术符号)选项。

(5) 出现 Select how the RTL/Tech Viewer behaves when it is initially invoked 界面，在该界面中选择 Start with a schematic of the top-level block 选项。

(6) 单击 OK 按钮。

(7) 如图 3.24 所示，显示出设计模块的顶层端口。

(8) 在图 3.24 内的界面内,双击顶层端口符号。

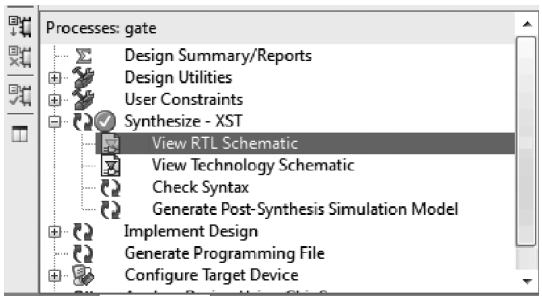


图 3.23 查看 RTL 原理图

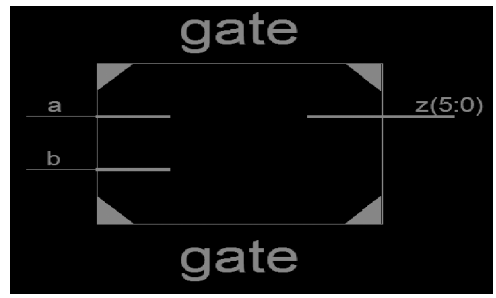


图 3.24 设计顶层端口

(9) 如图 3.25 所示,打开设计的内部结构图。

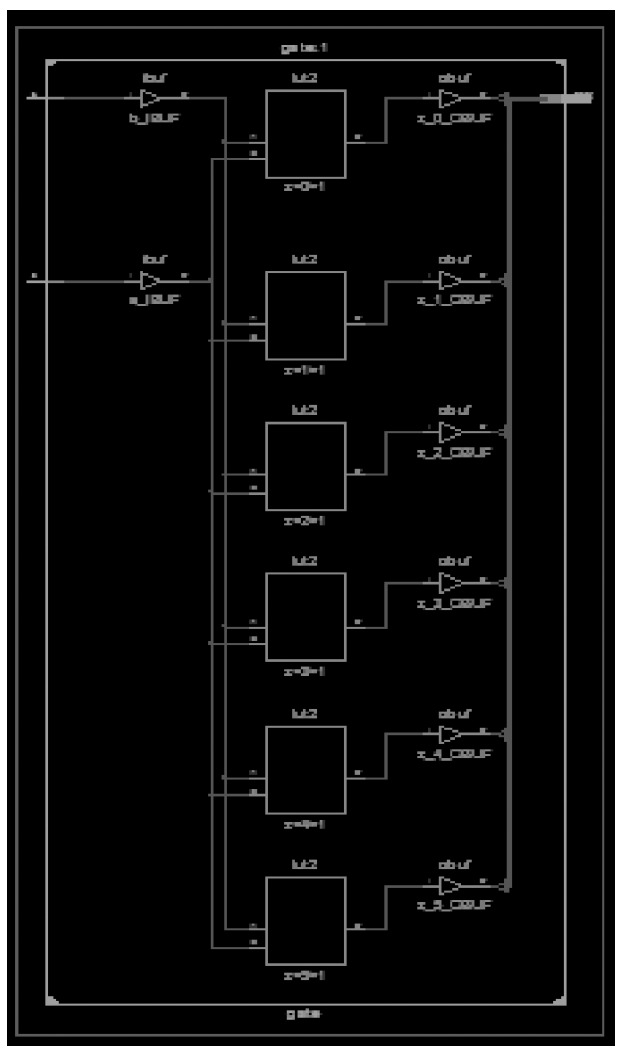


图 3.25 设计的内部结构

(10) 双击其中一个 LUT 图标,打开如图 3.26 所示界面。

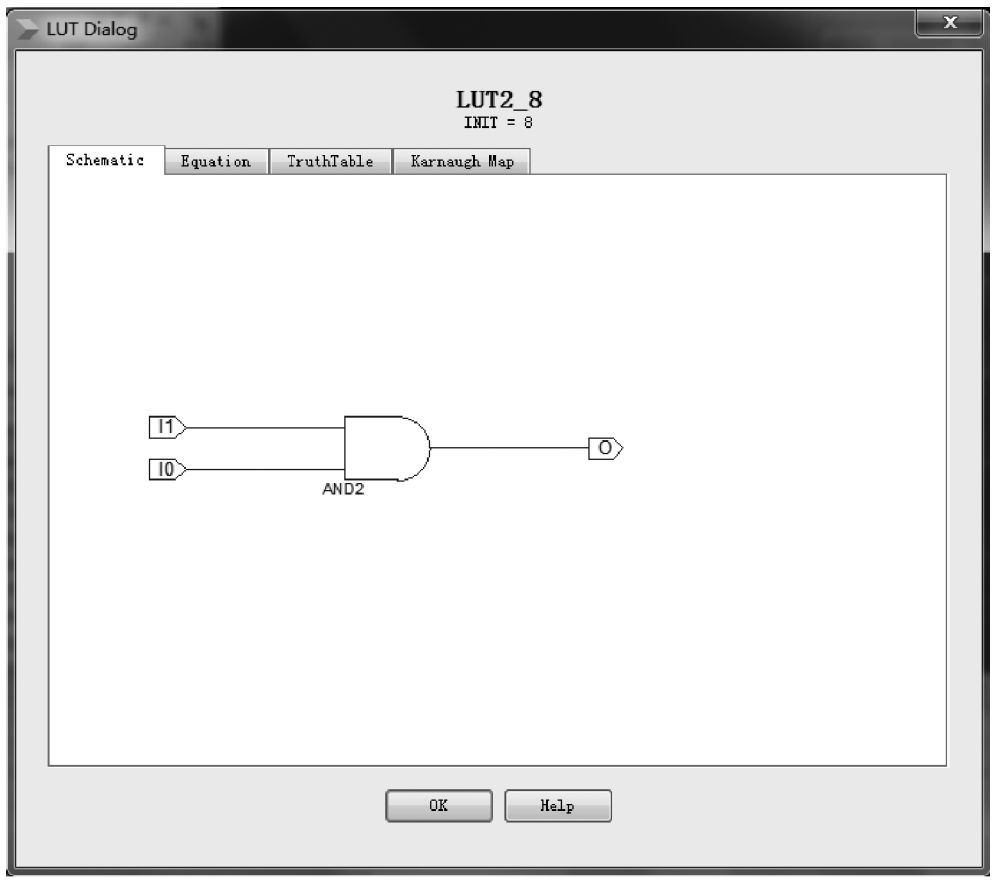


图 3.26　LUT 内部结构

(11) 在图 3.26 所示的界面内,单击 Equation 标签。

(12) 如图 3.27 所示,给出了该 LUT 内部逻辑关系的布尔逻辑表达式。

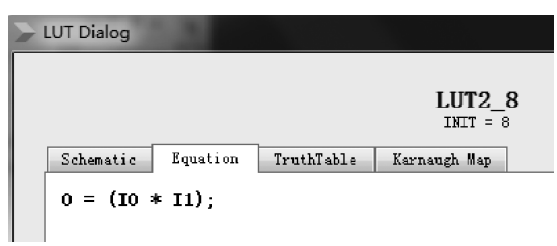


图 3.27　逻辑表达式

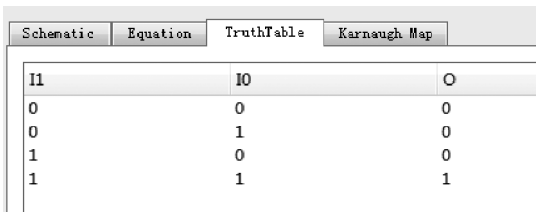
(13) 在图 3.27 所示的界面内,单击 TruthTable 标签。

(14) 如图 3.28 所示,给出了该 LUT 内部逻辑关系的真值表。

(15) 在图 3.28 所示的界面内,单击 Karnaugh Map 标签。

(16) 如图 3.29 所示,给出了该 LUT 内部逻辑关系的卡诺图的表示。

(17) 单击 OK 按钮,退出查看 LUT 内部结构界面。



I1	I0	O
0	0	0
0	1	0
1	0	0
1	1	1

图 3.28　逻辑关系的真值表描述

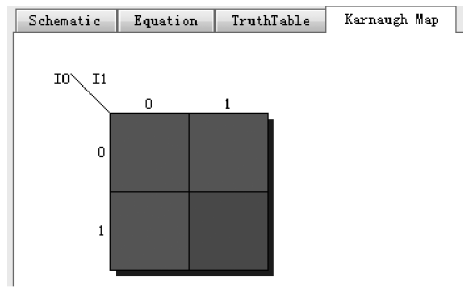


图 3.29　卡诺图表示

思考与练习 2　请说明综合的作用。

思考与练习 3　请观察说明该设计的整体结构。

(提示：FPGA 的原理是使用 LUT 实现组合逻辑功能的结构)。

思考与练习 4　深入理解 HDL 语言的作用。

思考与练习 5　读者可以打开其他 5 个 LUT，分析其内部结构。

3.7　设计行为仿真

如图 3.30 所示，测试平台以行为级描述为主，不使用寄存器传输级的描述形式。

在 Xilinx 高版本的 ISE 集成设计环境中，只提供了使用 HDL 语言生成测试向量的方法。下面介绍生成测试向量，并执行行为级仿真的步骤。如图 3.31 所示，在 Design 面板的 View 中，将单选按钮从 Implementation 改到 Simulation。这样，将设计流程从实现流程切换到仿真流程。

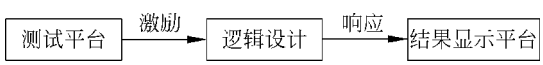


图 3.30　测试平台的作用

图 3.31　切换设计流程

3.7.1　为 Verilog HDL 设计添加测试向量

本节将添加 Verilog HDL 测试向量，下面给出添加 Verilog HDL 测试向量的步骤，其步骤主要包括：

(1) 如图 3.32 所示，选择 gate.v 文件，单击右键，出现浮动菜单，选择 New Source。

(2) 如图 3.33 所示，出现 New Source Wizard-Select Source Type 界面，在该界面内按如下参数设置：①在左侧选择 Verilog Test Fixture；②File name：test。

(3) 单击 Next 按钮。

(4) 如图 3.34 所示，出现 New Source-Associate Source(新源文件-关联源文件)界面，选择需要仿真的设计文件，在该设计中，只有一个设计文件 gate。如果一个设计中，有多个设计文件，则需要手工选择要仿真的设计文件。

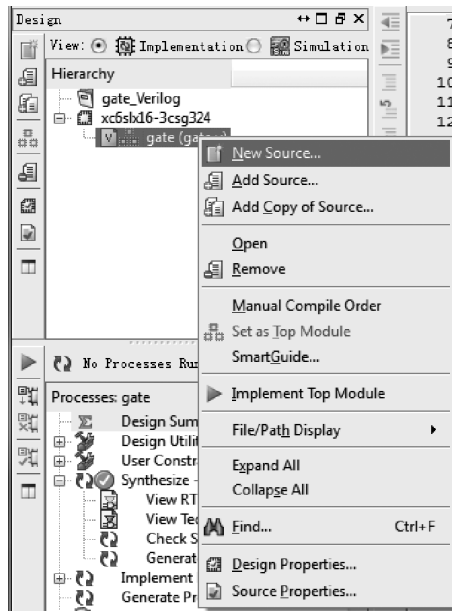


图 3.32　添加 gate.v 文件

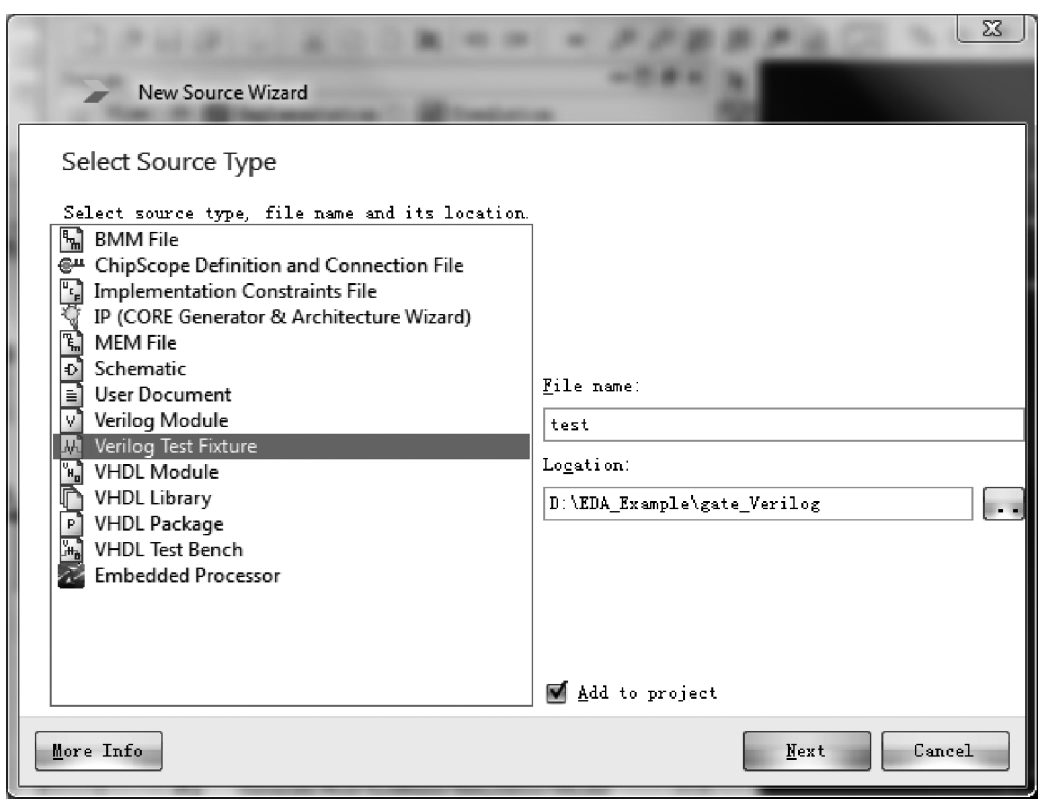


图 3.33　添加 test.v 仿真文件

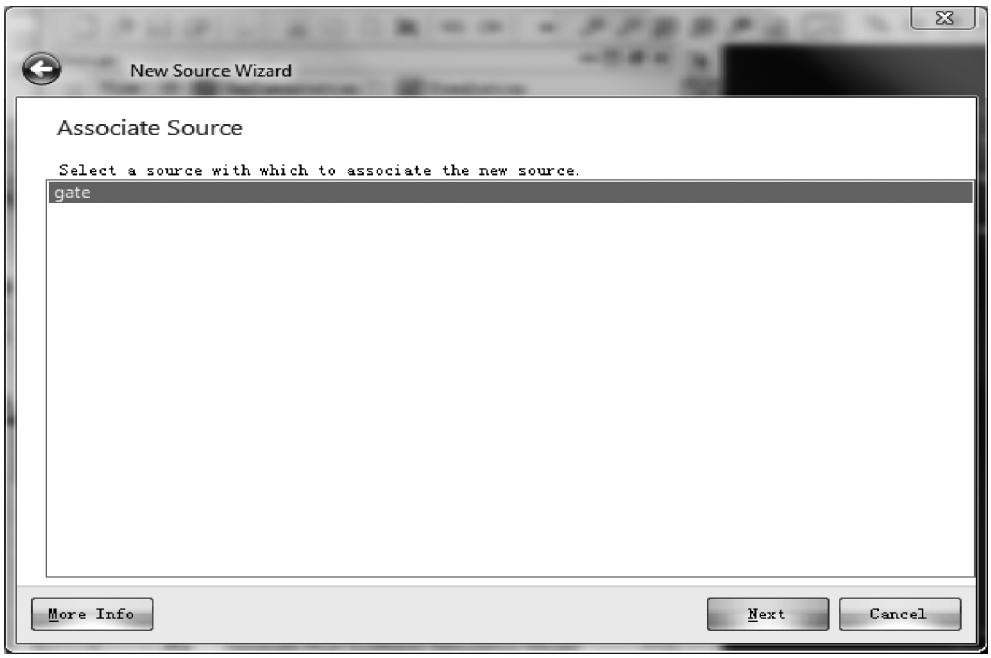


图 3.34　关联设计文件

(5) 单击 Next 按钮。

(6) 出现 New Source-Summary(新源文件-总结)界面,生成的仿真文件的名字为 test.v。

(7) 单击 Finish 按钮。

(8) 如图 3.35 所示,在 Hierarchy 窗口下,新添加了 test.v 文件,并且自动打开 test.v 文件。

(9) 如图 3.36 所示,在 test.v 文件中添加测试向量。

(10) 保存设计文件 test.v。

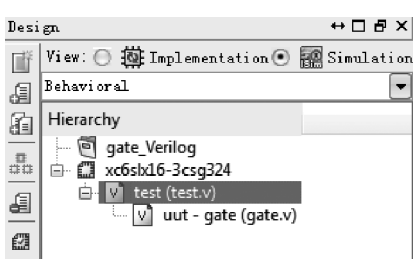


图 3.35　新添加了 test.v 文件

```
51
52      always
53      begin
54         a = 0;
55         b = 0;
56         #100;
57
58         a =0;
59         b =1;
60         #100;
61
62         a =1;
63         b =0;
64         #100;
65
66         a =1;
67         b =1;
68         #100;
69      end
70  endmodule
```

图 3.36　添加 verilog 测试向量

3.7.2　为 VHDL 设计添加测试向量

本节将添加 VHDL 测试向量，下面给出添加 VHDL 测试向量的步骤，其步骤主要包括：

(1) 类似地，选择 gate.vhd 文件，单击右键，出现浮动菜单，选择 New Source。

(2) 如图 3.33 所示，出现 New Source Wizard-Select Source Type 界面，在该界面内按如下参数设置：①在左侧选择 VHDL Test Bench；②File name：test。

(3) 单击 Next 按钮。

(4) 如图 3.34 所示，出现 New Source-Associate Source(新源文件-关联源文件)界面，选择需要仿真的设计文件，在该设计中，只有一个设计文件 gate。如果一个设计中，有多个设计文件，则需要手工选择要仿真的设计文件。

(5) 单击 Next 按钮。

(6) 出现 New Source-Summary(新源文件-总结)界面。生成的仿真文件的名字为 test.vhd。

(7) 单击 Finish 按钮。

(8) 如图 3.37 所示，在 Hierarchy 窗口下，新添加了 test.vhd 文件，并且自动打开 test.vhd 文件。

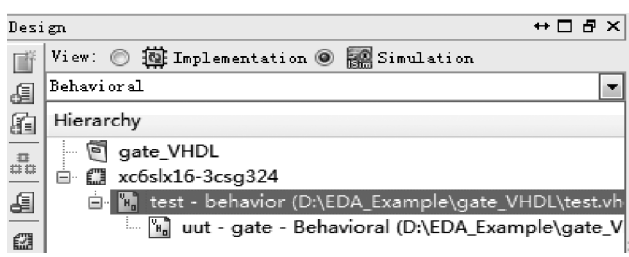


图 3.37　新添加了 test.vhd 文件

(9) 如图 3.38 所示，在 test.vhd 文件中注释掉第 61 行、第 71～92 行的内容。

```
59
60   --  constant <clock>_period : time := 10 ns;
61
```

```
71   --   -- Clock process definitions
72   --   <clock>_process :process
73   --   begin
74   --    <clock> <= '0';
75   --    wait for <clock>_period/2;
76   --    <clock> <= '1';
77   --    wait for <clock>_period/2;
78   --   end process;
79   --
80   --
81   --   -- Stimulus process
82   --   stim_proc: process
83   --   begin
84   --      -- hold reset state for 100 ns.
85   --      wait for 100 ns;
86   --
87   --      wait for <clock>_period*10;
88   --
89   --      -- insert stimulus here
90   --
91   --      wait;
92   --   end process;
```

图 3.38　注释掉和本设计无关的测试代码

(10) 如图 3.39 所示,在 test.vhd 文件中添加测试向量。

(11) 保存测试文件 test.vhd。

```
93      process
94      begin
95        a<='0';
96        b<='0';
97        wait for 100 ns;
98        a<='0';
99        b<='1';
100       wait for 100 ns;
101       a<='1';
102       b<='0';
103       wait for 100 ns;
104       a<='1';
105       b<='1';
106       wait for 100 ns;
107     end process;
108
109   END;
```

图 3.39 添加 VHDL 测试向量

3.7.3 运行行为仿真

本节将运行行为仿真,并分析行为仿真结果。下面给出运行行为仿真的步骤,其步骤主要包括:

(1) 启动 ISim 仿真程序。①对于 Verilog 设计者,如图 3.40 所示,选择 test.v 文件;②对于 VHDL 设计者,如图 3.41 所示,选择 test.vhd 文件。

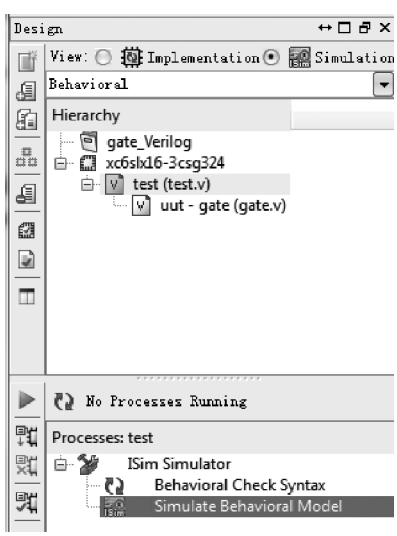


图 3.40 运行仿真 test.v

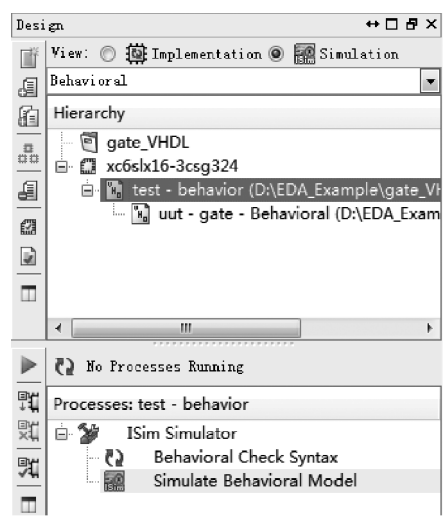


图 3.41 运行仿真 test.vhd

(2) 在图 3.40 或者图 3.41 内下方的处理子窗口中,选择并展开 Isim Simulator。

(3) 在展开项中,双击 Simulate Behavioral Model(仿真行为模型)。

(4) 出现如图 3.42 所示的界面。

(5) 在该界面的工具栏内,单击 按钮,对波形进行缩小直到出现如图 3.43 所示的波形。

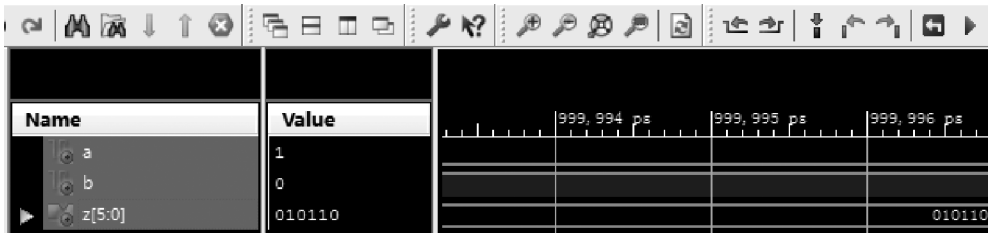


图 3.42　仿真波形界面 1

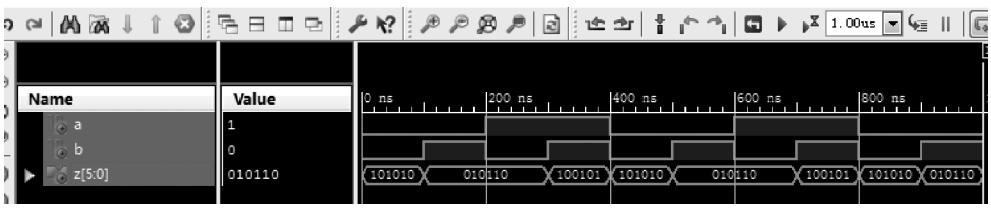


图 3.43　仿真波形界面 2

(6) 展开图 3.43 内的 z[5:0],出现如图 3.44 所示的波形界面。

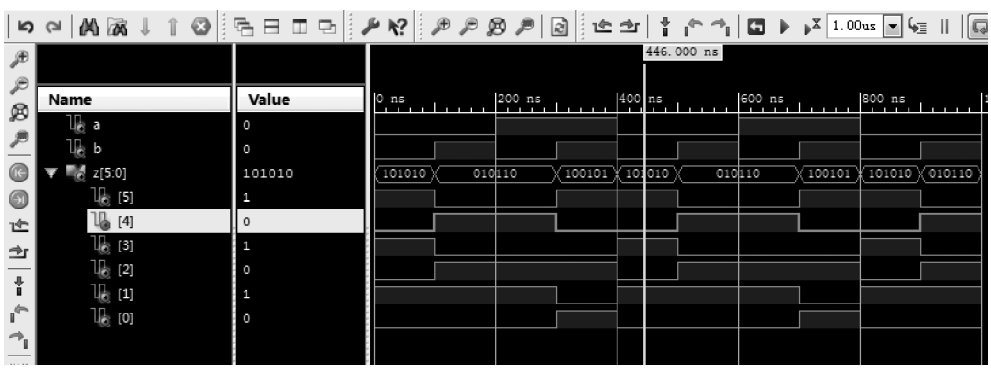


图 3.44　仿真波形界面 3

(7) 如图 3.45 所示,在仿真界面下方的 Console(控制台)界面内,可以通过输入不同的命令对仿真进行控制。

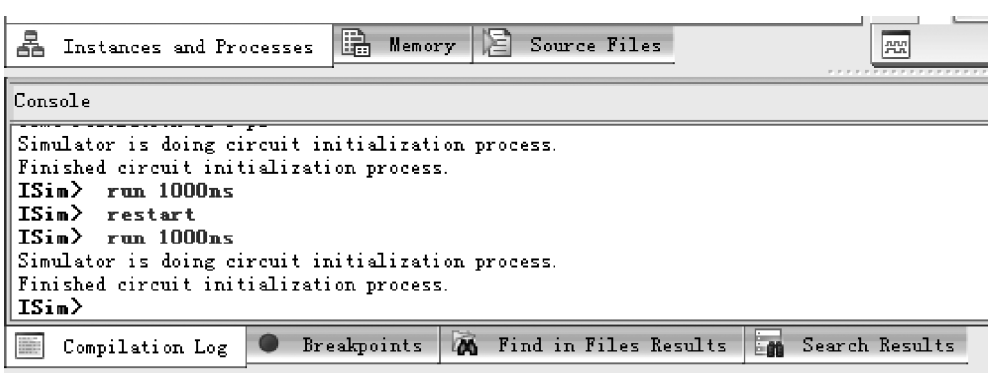


图 3.45　Console 界面

如果想详细了解 ISim 仿真工具的命令,在 Isim 后面输入 help 命令即可。

(8) 在 ISim 后面输入 quit 命令,退出仿真工具界面。

(9) 出现 Quit Simulation(退出仿真)对话框界面。

(10) 单击 Yes 按钮。

思考与练习 6 请说明进行仿真所需要的条件。

思考与练习 7 请说明行为仿真的功能。

3.8 添加引脚约束文件

本节将添加引脚约束文件。下面给出添加引脚约束文件的步骤,其步骤主要包括:

(1) 如图 3.46 所示,在 Design 面板下,将单选按钮,从 Simulation 切换到 Implementation。

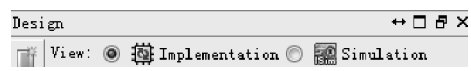


图 3.46 View 面板

(2) 在 Hierarchy 窗口下,选择器件名字 xc6slx16-3csg324。单击右键,出现浮动菜单。选择 New Source。

(3) 如图 3.47 所示,出现 New Source Wizard-Select Source Type 界面。在该界面内按如下参数设置:①选择源文件类型:Implementation Constraints File;②File name:gate。

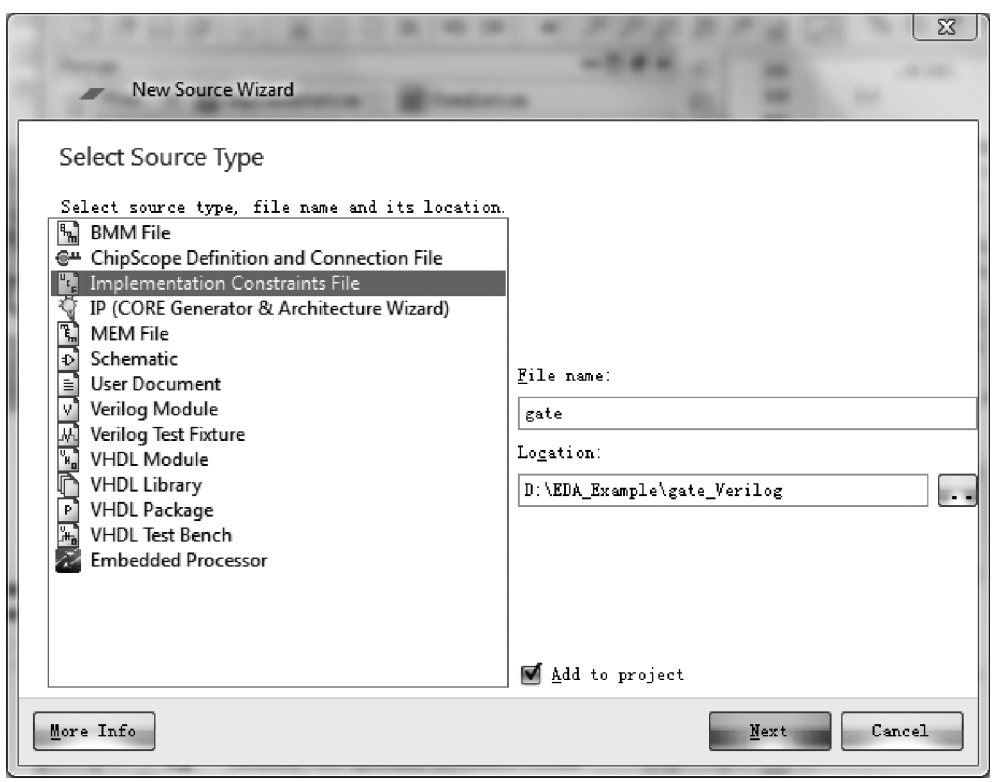
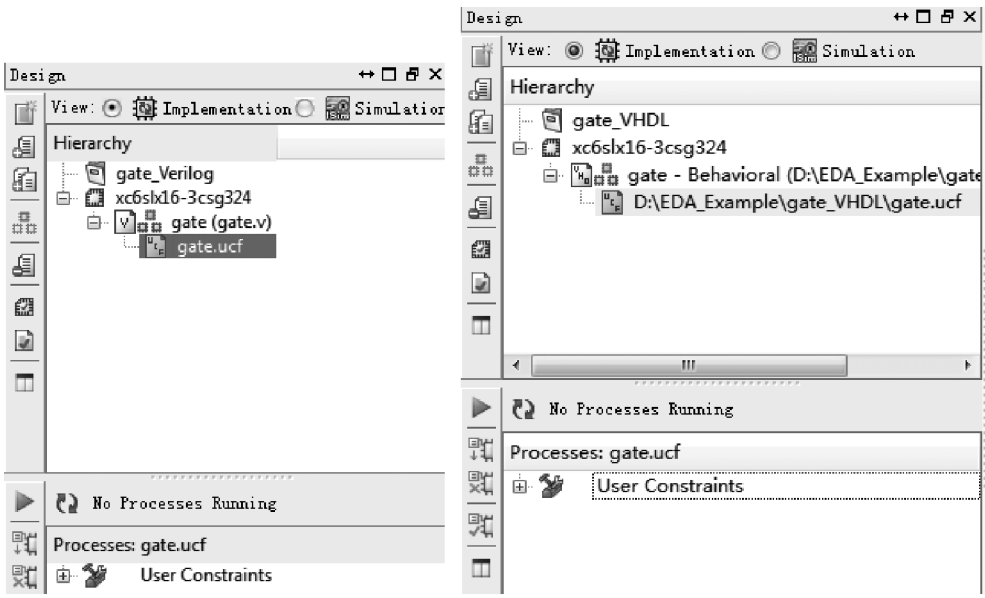


图 3.47 新建.UCF 文件

(4) 出现 New Source Wizard-summary 界面。新添加的用户约束文件为 gate. ucf。

(5) 单击 Finish 按钮。

(6) 如图 3.48 所示,在主界面的 Hierarchy 窗口下,添加 gate. ucf 文件。系统自动打开了空的 gate. ucf 文件,手动关闭该文件。



(a) Verilog 设计界面　　(b) VHDI设计界面

图 3.48　出现. ufc 界面

(7) 在图 3.48(a)所示的界面内选中 gate. v 文件。对于使用 VHDL 的读者,在图 3.48(b)所示的界面内选中 gate. vhd 文件。

(8) 如图 3.49 所示,在图 3.48(a)或者图 3.48(b)的下方窗口中,选择并展开 User Constraints。

(9) 在展开项中,选择并双击 I/O Pin Planning(PlanAhead)-Post-Synthesis。

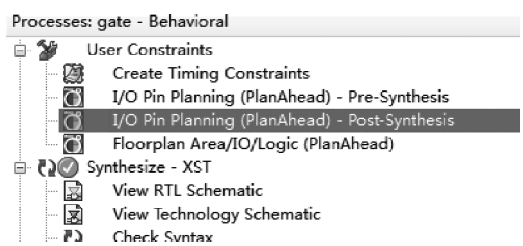


图 3.49　用户约束文件选择入口

(10) 如图 3.50 所示,出现提示对话框界面,询问是否继续打开 PlanAhead 设计界面。单击 Yes 按钮。

(11) 出现如图 3.51 所示界面,等待网表导入完成后,单击图中的 Close 按钮。

(12) 对引脚位置进行约束。如图 3.52 所示,展开 z 和 Scalar ports。①在 Site 栏下,对所对应行的引脚位置进行约束;②在 I/O Std 栏下,对所对应行的引脚电气标准进行约束。表 3.3 给出了所约束引脚的位置和对应的电气标准。

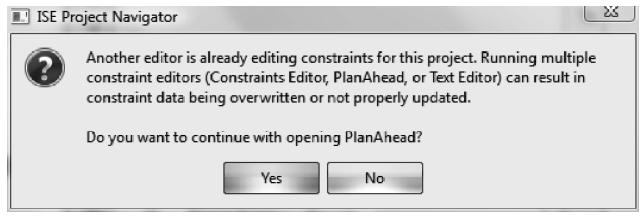


图 3.50　提示存在 ucf 对话框界面

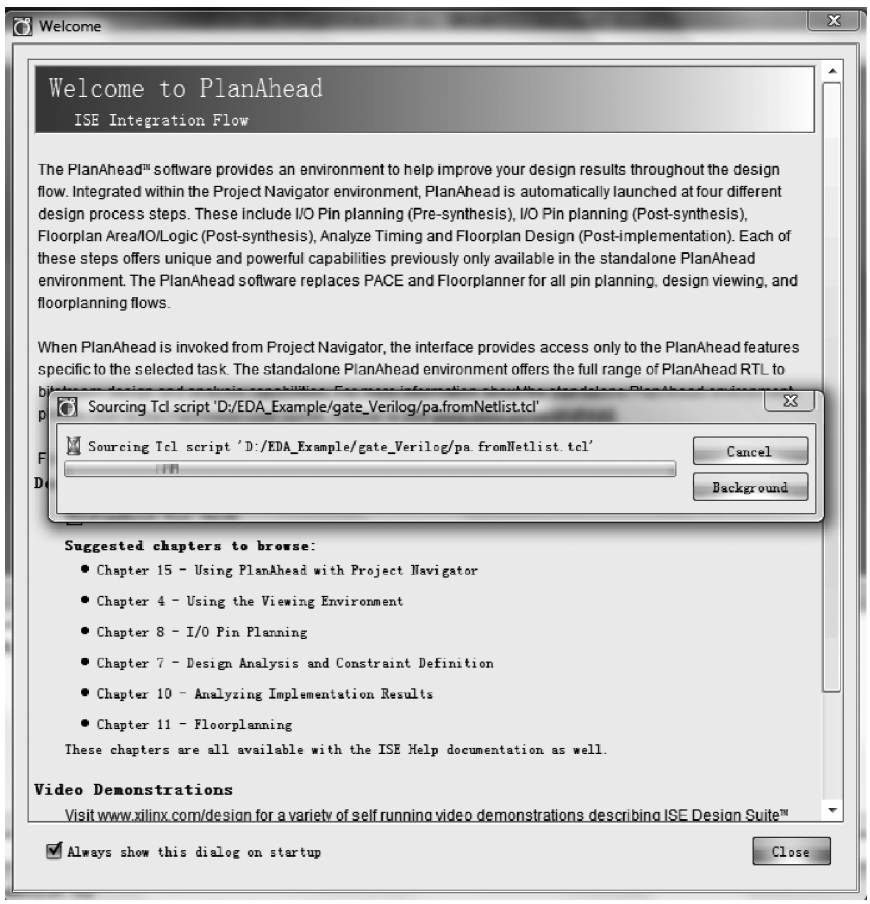


图 3.51　PlanAhead 工具的 Welcome 界面

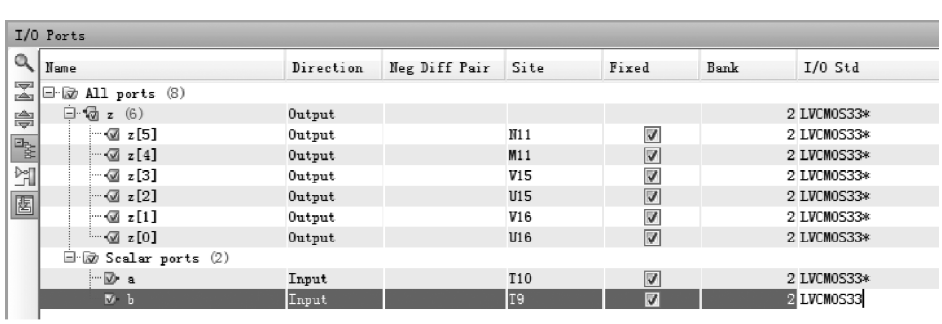


图 3.52　对应引脚约束

表 3.3　约束引脚的位置和电气标准

网络名	FPGA 引脚位置	电气标准
a	T10	LVCMOS33
b	T9	LVCMOS33
z[5]	N11	LVCMOS33
z[4]	M11	LVCMOS33
z[3]	V15	LVCMOS33
z[2]	U15	LVCMOS33
z[1]	V16	LVCMOS33
z[0]	U16	LVCMOS33

(13) 保存并退出约束编辑器界面。

思考与练习 8　如图 3.53 所示，单击 Package 标签，查看 FPGA I/O 分布特性、电气特性和分组管理特性。

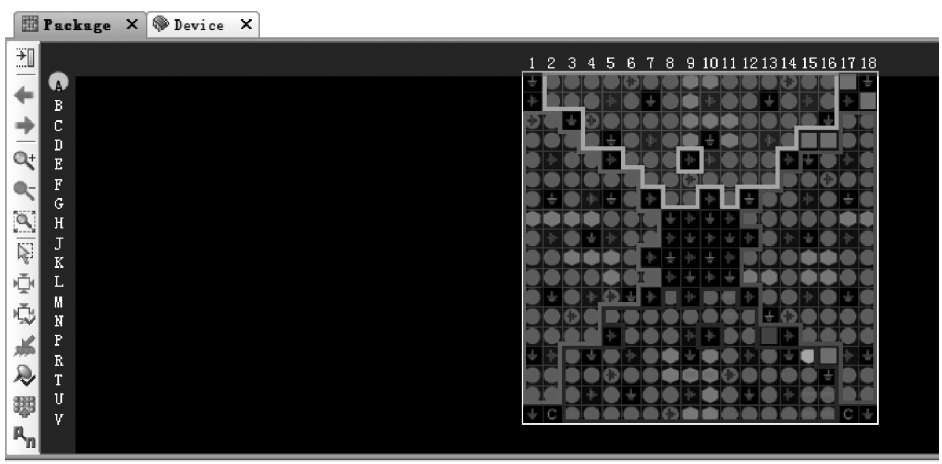


图 3.53　FPGA I/O 封装图

思考与练习 9　如图 3.54 所示，单击 Device 标签，查看 FPGA 内的片内逻辑资源。

图 3.54　FPGA 内部结构图

提示：在查看内部结构的时候，单击放大按钮。

(14) 双击图 3.48 Hierarcy 面板内的 gate.ucf 文件图标，查看或修改管脚约束文件。

(15) 如图 3.55 所示，给出了 gate.ucf 文件的文本描述格式。

```
1
2  # PlanAhead Generated physical constraints
3
4  NET "z[5]" LOC = N11;
5  NET "z[4]" LOC = M11;
6  NET "z[3]" LOC = V15;
7  NET "z[2]" LOC = U15;
8  NET "z[1]" LOC = V16;
9  NET "z[0]" LOC = U16;
10 NET "a" LOC = T15;
11 NET "b" LOC = T9;
12
13 # PlanAhead Generated IO constraints
14
15 NET "z[5]" IOSTANDARD = LVCMOS33;
16 NET "z[4]" IOSTANDARD = LVCMOS33;
17 NET "z[3]" IOSTANDARD = LVCMOS33;
18 NET "z[2]" IOSTANDARD = LVCMOS33;
19 NET "z[1]" IOSTANDARD = LVCMOS33;
20 NET "z[0]" IOSTANDARD = LVCMOS33;
21 NET "a" IOSTANDARD = LVCMOS33;
22
```

图 3.55　约束文件代码

(16) 关闭 gate.ucf 文件。

思考与练习 10　分析该约束文件，了解和掌握 Xilinx 约束文件的格式(该文件非常重要)。

3.9　设计实现

ISE 中的实现(Implement)过程，是将综合输出的逻辑网表翻译成所选器件的底层模块与硬件原语，将设计映射到器件结构上，并在所选的器件上进行布局布线，达到在选定器件上最终实现设计的目的。实现过程主要分为下面 3 个步骤：

1) Translate(翻译)

翻译的主要作用是将综合输出的逻辑网表翻译为 Xilinx 特定器件的底层结构和硬件原语。

2) Map(映射)

映射的主要作用是将设计映射到具体型号的器件上。

3) Place&Route(布局布线)

布局布线的主要作用是调用 Xilinx 布局布线器，根据用户约束和物理约束，对设计模块进行实际的布局，并根据设计连接，对布局后的模块进行布线，产生 FPGA 配置文件。

3.9.1　运行设计实现工具

本节将运行实现工具对设计进行实现。下面给出对设计进行实现的步骤，其步骤主要包括：

(1) 选择将要实现的文件。对于 Verilog HDL 的设计者,选择 gate. v 文件。对于 VHDL 的设计者,选择 gate. vhd 文件。

(2) 如图 3.56 所示,在处理子窗口下,选择并双击 Implement Design 选项。

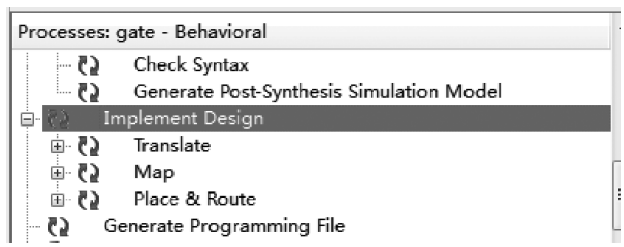


图 3.56　调用实现工具

(3) 开始运行实现工具。

(4) 如图 3.57,给出实现完成后的界面。

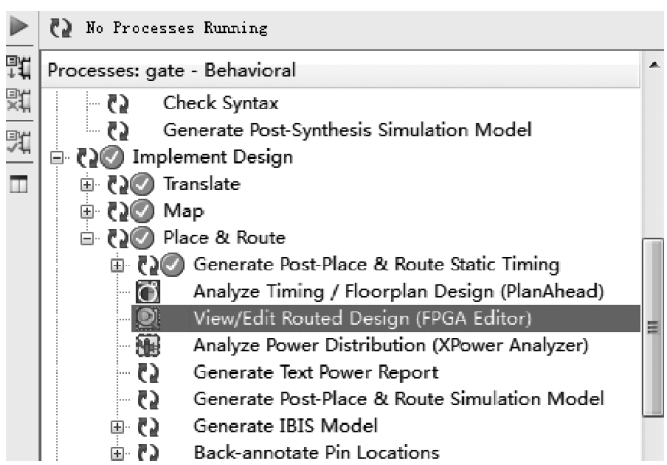


图 3.57　实现完成后的界面

3.9.2　查看布局布线结果

本节将查看布局布线后的结果。下面给出查看布局布线后结果的步骤,其步骤主要包括:

(1) 如图 3.57 所示,找到并展开 Implement Design 选项。

(2) 在展开项中找到 Place & Route 选项。

(3) 展开 Place & Route 选项。

(4) 在展开项中,找到并双击 View/Edit Routed Design(FPGA Editor)选项。

(5) 如图 3.58 所示,给出了 FPGA Editor 主界面。

(6) 单击图内的 按钮,放大视图。

(7) 如图 3.59 所示,找到图中标记为蓝色的位置。表示使用的引脚和内部的 slice 资源。绿线表示用于互联当前设计资源的连线。

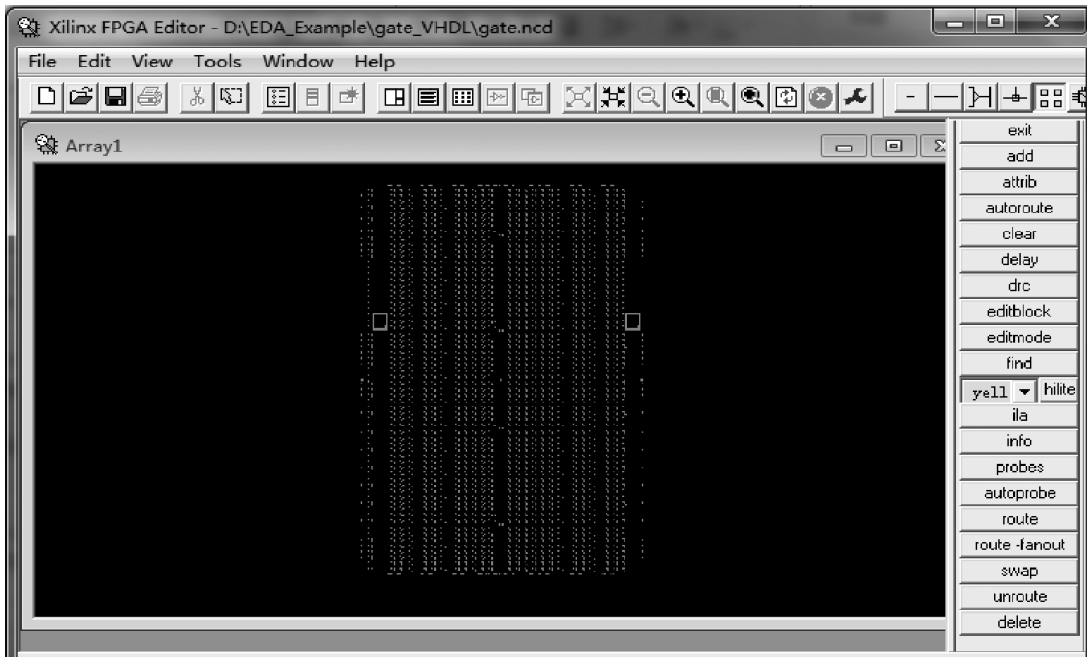


图 3.58　FPGA 主界面

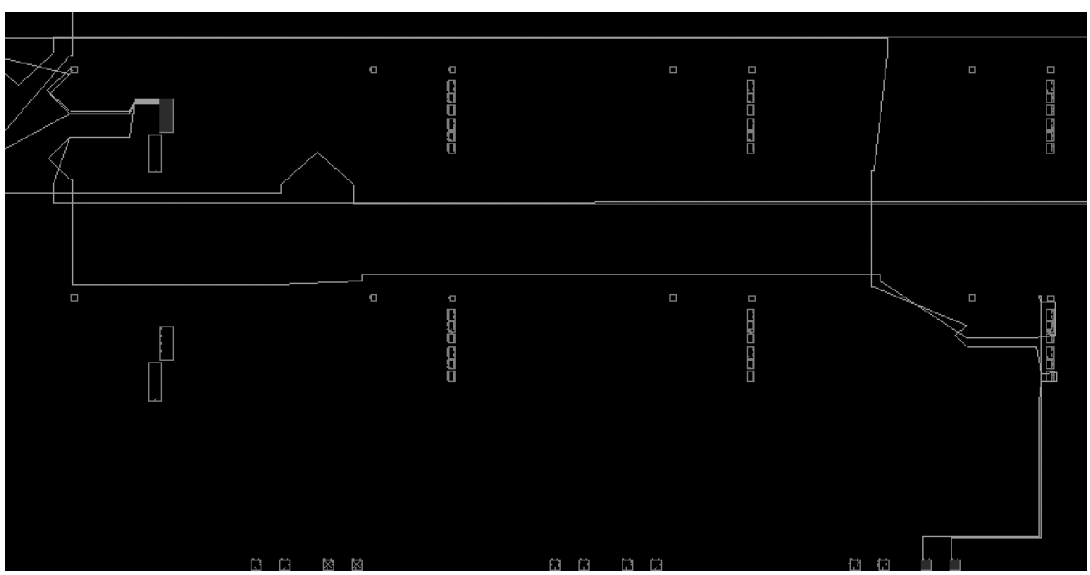


图 3.59　FPGA 内部互联结构

思考与练习 11　如图 3.60 所示，双击图中蓝色的 Slice，打开内部结构。如图 3.61 所示，根据第 2 章所介绍的 FPGA 的内部结构原理，对其结构进行分析。

(8) 退出 FPGA Editor 编辑器界面。

思考与练习 12　请说明设计实现所包含的主要过程以及每个过程实现的功能。

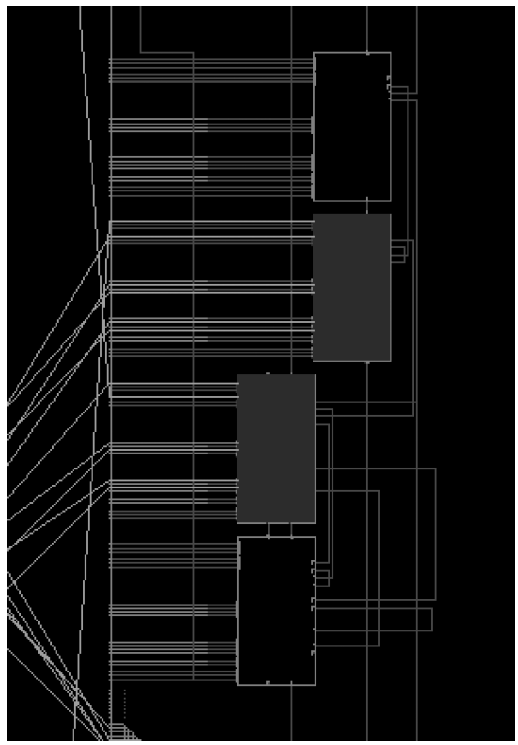


图 3.60　所使用的 Slice

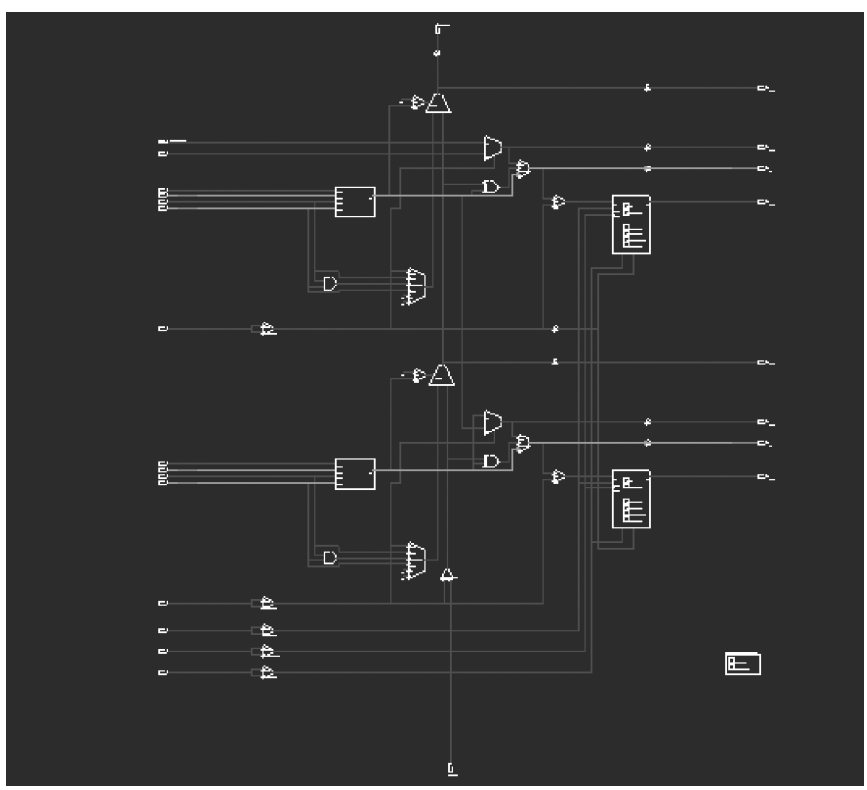


图 3.61　Slice 内部结构

3.10 布局布线后仿真

本节将执行布局布线后仿真,并分析时序仿真后的结果。下面给出执行时序仿真的步骤,其步骤主要包括:

(1) 如图 3.62 所示,在 Design 面板的 View 中,将单选按钮从 Implementation 改到 Simulation。这样,将设计流程从实现流程切换到仿真流程。

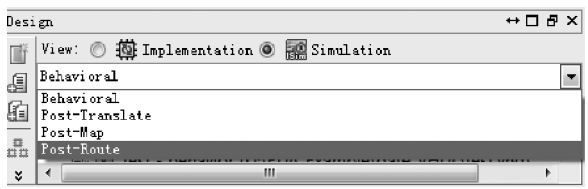


图 3.62 切换设计流程

(2) 在下拉框中,选择 Post-Route(布线后)选项。

(3) 选择 test.v(对于 Verilog HDL 设计者)或者 test.vhd 文件(对于 VHDL 设计者)。

(4) 如图 3.63 所示,在处理子窗口内选择并展开 Isim Simulator。

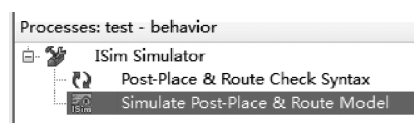


图 3.63 时序仿真入口

(5) 在展开项中,选择并双击 Simulate Post-Place & Route Model(仿真布局和布线后模型)选项。

(6) 打开 ISim 工具,调整波形窗口,使得波形进入观察窗口中。

(7) 如图 3.64 所示,查看图中白色画圈的地方。

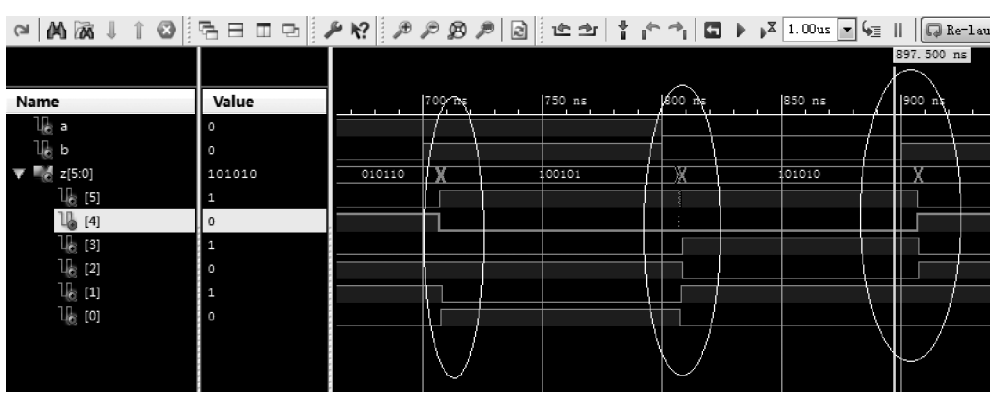


图 3.64 时序仿真波形

思考与练习 13 仔细观察和分析下面的情况。

① 输入和输出之间存在延迟,延迟在波形上面如何体现?以及形成延迟的原因。

② 不同输出之间的延迟并不相同,如何表现在波形上?以及原因。

③ 在 z[5:0]每个数据之间,存在毛刺,如何表现在波形上?以及原因。

④ 这些时序因素,如何影响在 FPGA 上所设计数字系统的工作速度?

(8) 退出 ISim 时序仿真界面。

思考与练习 14　请说明布局布线后仿真所实现的功能。

3.11　产生比特流文件

本节将生成比特流文件。下面给出生成比特流文件的步骤,其步骤主要包括:

(1) 在 View 面板中,将单选按钮从 Simulation 切换到 Implementation。

(2) 选择 gate.v 文件(对于 Verilog HDL 设计者)或者 gate.vhd 文件(对于 VHDL 设计者)。

(3) 如图 3.65 在处理子窗口中,双击 Generate Programming File(生成编程文件)。

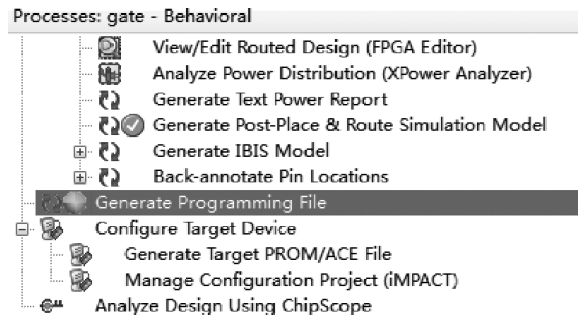


图 3.65　产生比特流选项

(4) 等待生成编程文件的过程结束。

3.12　下载比特流文件到 FPGA

JTAG 是最基本的一种用来配置 FPGA 的模式,通过它可以对 FPGA 进行在线调试。当使用 JTAG 完成对 FPGA 的调试后,根据不同 FPGA 的类型,需要选择使用外部的存储器保存配置文件代码(Spartan-3AN FPGA 采用了非易失性的 Flash 工艺,因此不需要外部 Flash 保存代码)。

下面给出使用 JTAG 实现对 Xilinx XUP 提供的 Nexys3 开发板进行配置的步骤。在将配置文件下载到 FPGA 器件前,必须将 USB 电缆连接到 PC 或笔记本电脑的 USB 接口,另一侧连接到 Nexys3 浇板卡的 J3(USB PROG)接口。并打开板上的 SW8 开关,给 Nexys3 浇板卡上电。其配置步骤如下:

(1) 在 Hierarchy 窗口下,选择 gate.v 文件(对于 Verilog HDL 设计者)或者 gate.vhd 文件(对于 VHDL 设计者)。

(2) 如图 3.66 所示,在 Processes 窗口中,找到并展开 Configure Target Deivce(配置目标器件)选项。在展开项中,找到并双击 Manage Configuration Project(iMPACT)(管理配置工程(iMPACT))选项。

(3) 如图 3.67 所示,打开 iMPACT 工具主界面,出现 Feedback Request 界面。

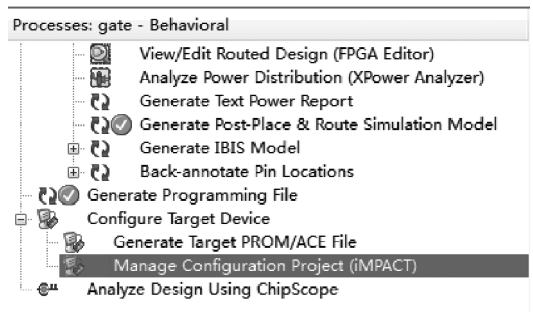


图 3.66　下载比特流到 FPGA 入口

图 3.67　调查对话框窗口界面

(4) 单击 No 按钮。如果没有出现图 3.67 的对话框界面,则跳过(3)和(4)步,直接到(5)步。

(5) 如图 3.68 所示,在 iMPACT Flows 面板窗口下,选中并双击 Boundary Scan(边界扫描)选项,在右侧出现空白窗口界面。

图 3.68　iMPACT 设计流程界面

(6) 如图 3.69 所示,在该空白界面下,单击右键,出现浮动菜单,选择 Initialize Chain(初始化 JTAG 链)选项。

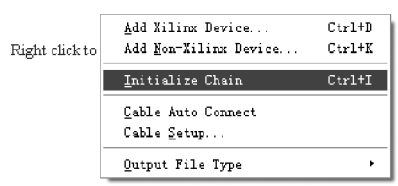


图 3.69　初始化 JTAG 链选项

(7) 出现进度条,扫描 JTAG 链路上的器件。当 JTAG 扫描成功时,出现图 3.70 所示的界面。从图中可以看到设计所选择的器件 xc6slx16 出现在扫描链路中,表示扫描数据由 TDI 输入,TDO 输出,并且提示 Identify Succeeded(识别成功)标志。

如果没有扫描到任何器件,请仔细检查硬件连接,并且确认开发平台正确上电。

(8) 弹出 Auto Assign Configuration Files Query Dialog(自动分配配置文件查询对话框)界面。单击 Yes 按钮,使用自动分配方式。

(9) 出现如图 3.71 所示的 Assign New Configuratior File(分配新配置文件)对话框界面,对于 VHDL 的设计者定位到下面目录

```
D:\EDA_Example\gate_VHDL
```

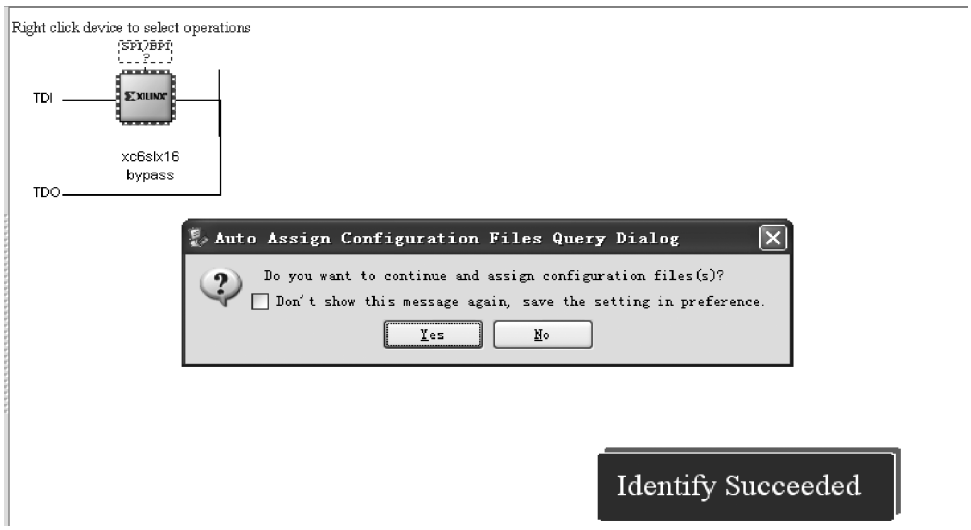


图 3.70　JTAG 链路扫描 FPGA 器件

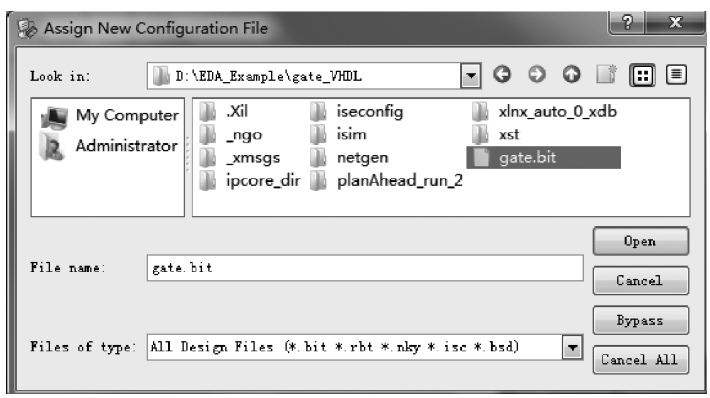


图 3.71　打开比特流文件

对于 Verilog HDL 的设计者定位到下面目录

```
D:\EDA_Example\gate_Verilog
```

(10) 在对应的目录下,选择 gate.bit 文件,该文件是前面所生成的 FPGA 配置文件该文件用于对 FPGA 进行配置。

(11) 单击 Open 按钮,为 FPGA 分配该配置文件。

(12) 如图 3.72 所示,出现 Attach SPI or BPI PROM(添加 SPI 或者 BPI PROM)对话框界面,该对话框界面询问是否为该 FPGA 分配 SPI 或者 BPI PROM。单击 No 按钮。这是因为现在使用的是 JTAG 模式,不使用 SPI 或 BPI PROM 存储器配置 FPGA。

图 3.72　添加 SPI 或 BPI PROM 对话框

(13) 如图 3.73 所示,出现 Device Programing Properties(器件编程属性)对话框界面,在该界面内,单击 OK 按钮。

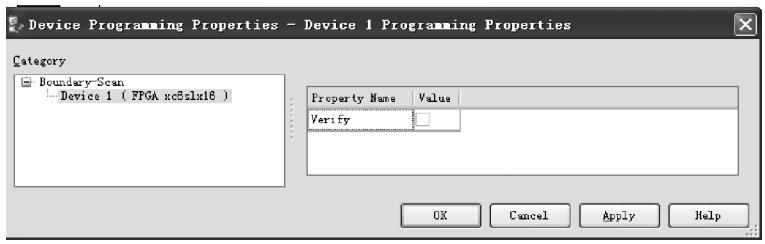


图 3.73 编程属性对话框

(14) 如图 3.74 所示,选中当前 FPGA 器件图标,单击右键,出现浮动菜单,选择 Program(编程)选项,开始对 FPGA 进行编程和配置。

(15) 如图 3.75 所示,出现 Configuration Operation Status(配置操作状态)对话框界面,表示了编程的进度。

(16) 编程结束后,出现图 3.76 所示的编程成功的提示界面。

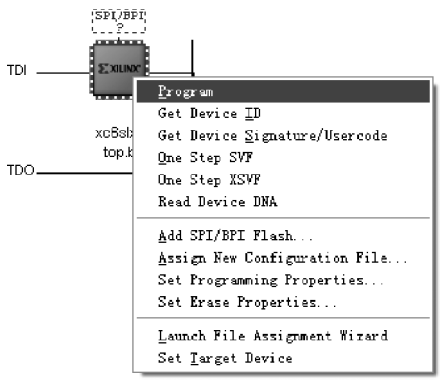


图 3.74 选择编程选项

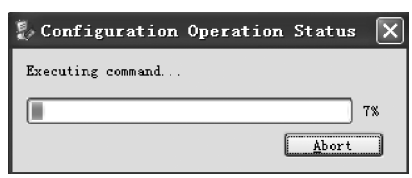


图 3.75 进度条界面

图 3.76 编程成功提示

(17) 拨动 Nexys3 浇开发版上的 SW0 和 SW1 开关,观察灯的状态变化。对 HDL 设计进行验证是否符合逻辑对应关系。

(18) 至此,完成了代码设计、综合、仿真、实现、编程文件生成和下载的所有设计流程。

(19) 不保存任何设计,退出 iMPACT 工具。

思考与练习 15 请说明所生成比特流文件的作用。

3.13 生成存储器配置文件并烧写存储器

本节将介绍生成存储器配置文件,并将存储器配置文件烧写到存储器中的方法。

3.13.1 生成 BPI 存储器配置文件

下面将详细介绍使用 BPI 模式配置器件的步骤,其步骤主要包括:

(1) 在 Hierarchy 面板窗口下,选择 gate.v 文件(对于 Verilog HDL 设计者)或者 gate.vhd

文件(对于 VHDL 设计者)。

(2) 如图 3.77 所示,在 Processes 窗口下,找到 Configure Target Device(配置目标器件)选项,并展开。在展开项中,选择并双击 Generate Target PROM/ACE File(生成目标 PROM/ACE 文件)选项。

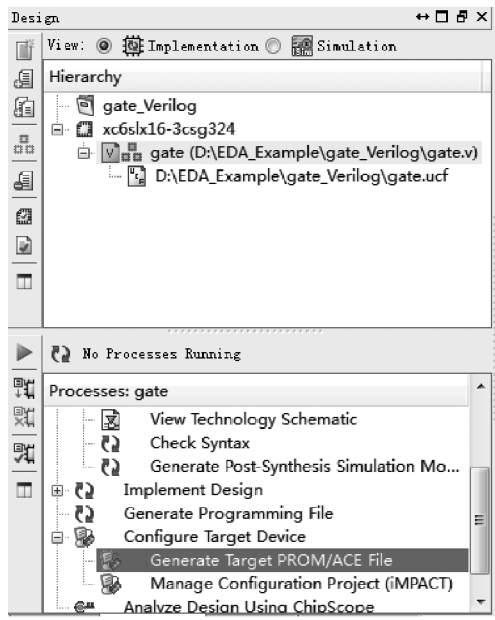


图 3.77　进入生成 PROM 文件界面入口

(3) 出现 Warning 窗口,单击 OK 按钮。

(4) 出现 Feedback Request 对话框窗口,单击 No 按钮。

(5) 如图 3.78 所示,在 iMPACT Flows 窗口下,选择 Create PROM File(PROM File Formatter)(创建 PROM 文件(PROM 文件格式器))选项,并双击该选项,打开生成 PROM 文件窗口界面。

(6) 如图 3.79 所示,在 Step 1. Select Storage Target(步骤一. 选择存储目标)面板窗口下,找到并展开 BPI Flash 选项。

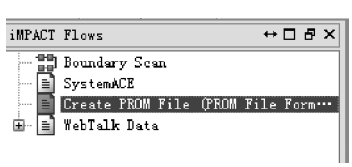


图 3.78　打开配置 PROM 文件界面选项

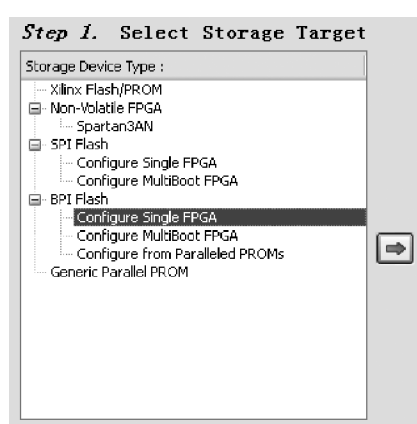


图 3.79　生成 BPI Flash 文件第一步操作

(7) 在展开项中,选择 Configure Single FPGA(配置单个 FPGA)选项,然后单击 按钮。

(8) 如图 3.80 所示,在 Step 2. Add Storage Device(s)(步骤二. 添加存储设备)面板窗口下,依次完成下面的步骤:①在 Target FGPA 右侧的下拉框中选择 Spartan-6,作为要配置的目标器件;②在 Storage Device(Bytes)选择 16M(单位是字节)(Nexys3 板卡上配置了 128Mb 的 RAM 存储器);③单击 Add Storage Device 按钮,然后在窗口中出现 16M 的标记;④然后单击 按钮,准备进入第三步操作。

(9) 如图 3.81 所示,依次完成下面的步骤:①为将要生成的 BPI 配置文件命名:BPI_DATA。②为将来所要生成的 BPI_DATA,指定存放的路径,在此将该文件保存到当前的工程目录下:对于 Verilog HDL 设计来说,目录为 D:\EDA_Example\gate_Verilog;对于 VHDL 设计来说,目录为 D:\EDA_Example\gate_VHDL;在 Data Width(数据宽度)右侧,通过下拉框,将其指定为(x16),表示数据宽度是 16 位的。

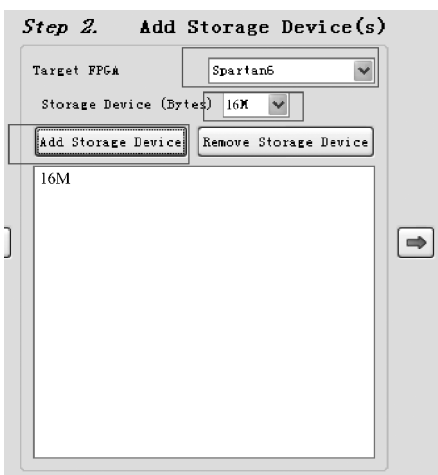


图 3.80 生成 BPI Flash 文件第二步操作

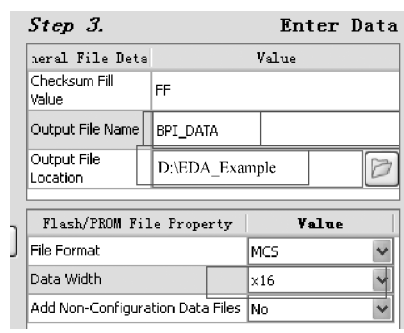


图 3.81 生成 BPI Flash 文件第三步操作

更详细的资料可以查看本书附录所提供的 NEXYS 的原理图,该设计中所用 FLASH 为 16 位宽度,128Mb 容量。

(10) 单击 OK 按钮。

(11) 如图 3.82 所示,出现 Add Device 对话框界面,单击 OK 按钮。

图 3.82 添加器件对话框

(12) 出现选择需要选换的比特流文件对话框界面。①定位到当前的工作目录下:对于 Verilog HDL 设计来说,目录为 D:\EDA_Example\gate_Verilog;对于 VHDL 设计来说,目录为 D:\EDA_Example\gate_VHDL。②找到 gate.bit,表示该二进制配置文件将被进行格式转换,将来编程到 BPI Flash。③单击 Open 按钮。

(13) 如图 3.83 所示,出现 Add Device 对话框界面,询问是否添加另一个器件的文件,单击 No 按钮,表示不添加其他文件。

(14) 弹出一个对话框界面,单击 OK 按钮。

(15) 弹出 MultiBoot BPI Revision and Data File Assignment 对话框界面,单击 OK 按钮。

(16) 如图 3.84 所示,在主界面左下角的 iMPACT Processes 窗口下,找到并双击 Generate File 选项。

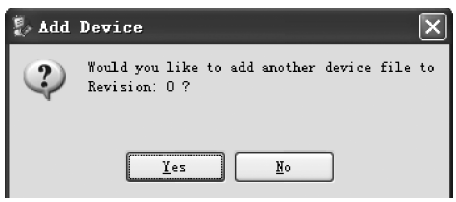


图 3.83　添加其他文件对话框

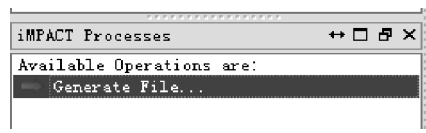


图 3.84　生成文件选项

(17) 在主界面右侧窗口,出现 Generate Succeeded 提示信息,表示成功生成 BPI_DATA.mcs 文件。

(18) 不保存任何信息,退出 iMPACT 工具。

3.13.2　编程 BPI 文件到 BPI 存储器

下面给出编程.mcs 文件到 BPI Flash 存储器的步骤,其步骤主要包括:

(1) 再次进入 iMPACT 工具。在 iMPACT 主界面内的 iMPACT Flows 面板下,选择并双击 Boundary Scan 选项。

(2) 在右侧窗口中,单击右键,出现浮动菜单,选择 Initialize Chain(初始化 JTAG 链)选项。

(3) 当出现 Auto Assign Configuration Files Query Dialog(自动分配配置文件对话框)界面时,单击 No 按钮。

(4) 然后单击 OK 按钮。

(5) 如图 3.85 所示,在 SPI/BPI 字符区域内,单击右键,出现浮动菜单。在浮动菜单内,选择 Add SPI/BPI Flash(添加 SPI/BPI Flash)选项。

(6) 定位到前面所指定的输出 BPI_DATA 文件的目录,然后在该目录下找到 BPI_DATA.mcs 文件。①对于 Verilog HDL 设计来说,目录为 D:\EDA_Example\gate_Verilog。②对于 VHDL 设计来说,目录为 D:\EDA_Example\gate_VHDL。

(7) 单击打开按钮。

(8) 如图 3.86 所示,从下拉菜单选择“28F128P30”(和 Nexys3 开发平台上所用的 BPI Flash 兼容)。单击 OK 按钮。

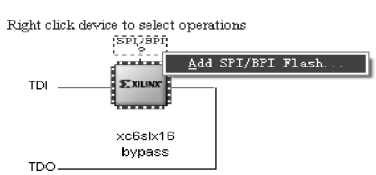


图 3.85　准备添加 BPI 文件

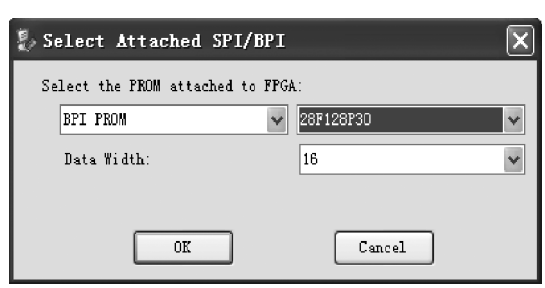


图 3.86　选择 BPI 存储器类型

(9) 如图 3.87 所示,选择标记为 FLASH 的区域,鼠标右键单击该区域,出现浮动菜单。在浮动菜单内,选择 Program 选项。

(10) 出现编程对话框界面。在该对话框界面内单击 OK 按钮,接受编程属性设置。

(11) 如图 3.88 所示,出现 BPI Flash 编程进度界面,当编程结束后,出现 Program Succeeded(编程成功)提示。

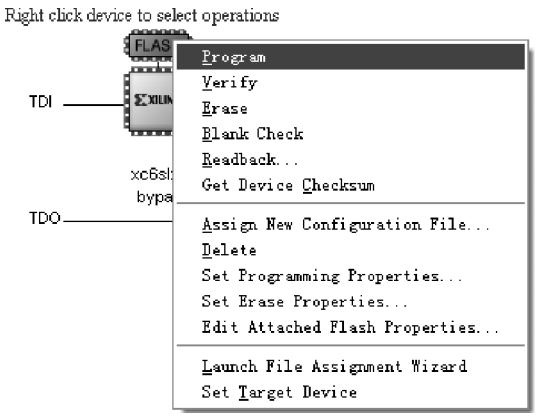


图 3.87 编程 BPI Flash 界面

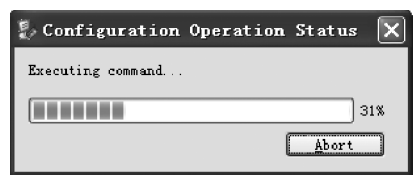


图 3.88 编程进度提示

(12) 退出 ISE 设计界面。

思考与练习 16 请查阅 Xilinx 手册,说明 Spartan-6 FPGA 所支持的下载模式。提示:支持的下载模式包括 JTAG 模式、串行模式、SelectMAP 模式、主 SPI 模式和主 BPI 模式。