

ISE 4.1i 快速入门



The Programmable Logic Company



技术领先的合作伙伴

目 录

| | |
|-------------------------------------|----|
| ISE 教程 | 3 |
| 教程概述 | 4 |
| 设计开始 | 5 |
| 软件要求 | 5 |
| 运行 ISE 软件 | 5 |
| 使用在线帮助 | 5 |
| 设计输入 (VHDL) | 6 |
| 创建一个新的工程项 | 6 |
| 创建一个计数器模块 | 6 |
| 利用计数器模板修改计数器模块 | 7 |
| 仿真行为模型 (功能仿真) | 9 |
| 创建一个 testbench 波形源文件 | 9 |
| 初始化计数器输入 | 10 |
| 生成预期的输出响应 | 10 |
| 使用 ModelSim 进行仿真 | 11 |
| 行为仿真 | 11 |
| 布局布线后的仿真 | 12 |
| 设计输入 (顶层为原理图) | 14 |
| 创建 VHDL 模块, 生成一个原理图符号 | 14 |
| 创建一个新的顶层原理图 | 14 |
| 例化 VHDL 模块 | 14 |
| 原理图中连线 | 15 |
| 为连线添加网络名 | 16 |
| 为总线添加网络名 | 17 |
| 添加输入/输出管脚标记 | 18 |
| 设计实现 | 20 |
| 运行实现设计 | 20 |
| 在资源分配器 (Floorplanner) 中查看设计布局 | 20 |
| 对顶层文件进行仿真 (时序仿真) | 22 |
| 创建一个 testbench 波形源文件 | 22 |
| 初始化计数器输入 | 22 |
| 生成预期的输出响应 | 22 |
| 布局布线后的仿真 | 24 |
| 附录: EDIF 设计方法 | 26 |
| 设计输入 | 26 |
| 创建一个新的工程项 | 26 |
| 添加 EDIF 源文件 | 27 |
| 设计实现 | 27 |
| 运行实现设计 | 27 |
| 在 FPGA Editor 中查看设计 | 28 |

ISE 教程

ISE – Integrated Software Environment

本教程主要是向 ISE 的初学者描述和演示，在 XILINX 的 ISE 集成软件环境中，如何用 VHDL 和原理图的方式进行设计输入；如何用 ModelSim 仿真工具对设计进行功能仿真和时序仿真；如何实现设计。

注：本教程的例子是在 PC 机上的 ISE 4. x 环境中实现的。

本教程包括以下几个章节：

- 教程概述
- 设计开始
- 设计输入 (VHDL)
- 行为模型仿真（功能仿真）
- 设计输入 (顶层为原理图)
- 设计实现
- 对顶层文件进行仿真（时序仿真）

要想学习如何用网表文件进行设计，请参考“附录：EDIF 设计方法”。

要想更深入地学习、了解 ISE 设计工具，请参阅 XILINX 网站上的 ISE 深入教程 (<http://www.support.xilinx.com/support/techsup/tutorials/>)。

教程概述

完成本教程后，你将会对以下的设计流程有一个清楚的认识：

- 创建一个新的工程项，用 Virtex 器件作为设计的目标器件；
- 用 ISE 的语言模板生成一个 4 位计数器的 VHDL 模块；
- 创建一个用波形输入的测试向量文件，并对 4 位计数器进行功能仿真；
- 创建一个顶层为原理图输入的设计；
- 在顶层原理图中调用计数器的 VHDL 模块；
- 把所有的模块连接在一起，并标出内连网线，总线以及 I/O 管脚的名称；
- 对 4 位计数器模块和顶层原理图的输入信号波形进行初始化设置，并对仿真长度进行约束；
- 对 4 位计数器模块进行功能和时序仿真，对顶层原理图设计进行时序仿真；
- 使用 Floorplanner 工具，查看经过布局布线后该设计的布局情况；
- 参照“附录：EDIF 设计方法”，完成网表(EDIF)文件输入的设计流程；
- 参照“附录：EDIF 设计方法”，使用 FPGA Editor，查看经过布局布线后该设计的布局和布线情况。

设计开始

本节主要描述本教程的软件要求，如何在 PC 机上运行 ISE 软件，以及如何获得在线帮助。

软件要求

为了完成本教程，你必须要在你的 PC 机上安装以下软件：

- ISE 4. x
- ModelSim VHDL

如果在XILINX的软件安装过程中遇到问题，请参阅《ISE安装指南》（*ISE Release Notes and Installation Guide*）。

软件安装完后，环境变量应如下所述：

1. 在 Windows 98 中，如果你的 ISE 软件安装在 C:\XILINX 目录下，ModelSim 软件安装在 C:\Modeltech 目录下，在 C:\autoexec.bat 文件中，应该包含如下设置：

```
SET XILINX=C:\XILINX
SET PATH=%XILINX%\BIN\NT;C:\Modeltech\win32;
SET LM_LICENSE_FILE=%XILINX%\data\license.dat;C:\Modeltech\win32\license.dat;
```

2. 在 Windows NT/2000 中，通过选择 开始 -> 设置 -> 控制面板 -> 系统 -> 高级 -> 环境变量 等选项来设置相应的环境变量。

关于在线申请 ModelSim XE Starter License，请选择 开始 -> 程序 -> ModelSim XE -> Submit License Request，可以在线申请 ModelSim Xilinx Edition Starter 的 license 文件。

运行 ISE 软件

对于 PC 机用户，可选择 开始 -> 程序 -> Xilinx ISE 4. x -> Project Navigator 来启动 ISE 软件。

使用在线帮助

你可以在 ISE 运行过程中，随时获得在线帮助。当你按下 F1 键时，帮助系统就会显示出正在运行过程的在线帮助。比如说，当你在运行 ECS(原理图输入)时按下 F1 键，那么关于如何使用 ECS 的在线帮助就会显示出来。

设计输入 (VHDL)

本节中，你将会利用 VHDL 语言模板来设计一个 4 位计数器模块。首先，创建一个新的工程项和生成一个计数器模块；然后，参照计数器 VHDL 语言模板来完成一个 4 位计数器的 VHDL 设计。

创建一个新的工程项

按照以下步骤创建一个新工程项：

1. 选择 File -> New Project ；
2. 在 New Project 对话框中的 Project Location 下，键入新工程项存放的路径，或者点击 Project Location 旁的浏览按钮，选择你想存放新工程项的路径；
3. 在 Project Name 下，键入 ‘Tutorial’。当你在 Project Name 下键入 ‘Tutorial’ 后，在 Project Location 下会自动创建一个 ‘Tutorial’ 的子目录；
4. 使用 Value 处的下拉菜单，你可以对每种属性进行选择。在下拉菜单列表中，点击所需的属性值如下：
 - 器件系列 (Device Family) : Virtex
 - 器件 (Device) : xcv50-6bg256
 - 设计流程 (Design Flow) : XST VHDL

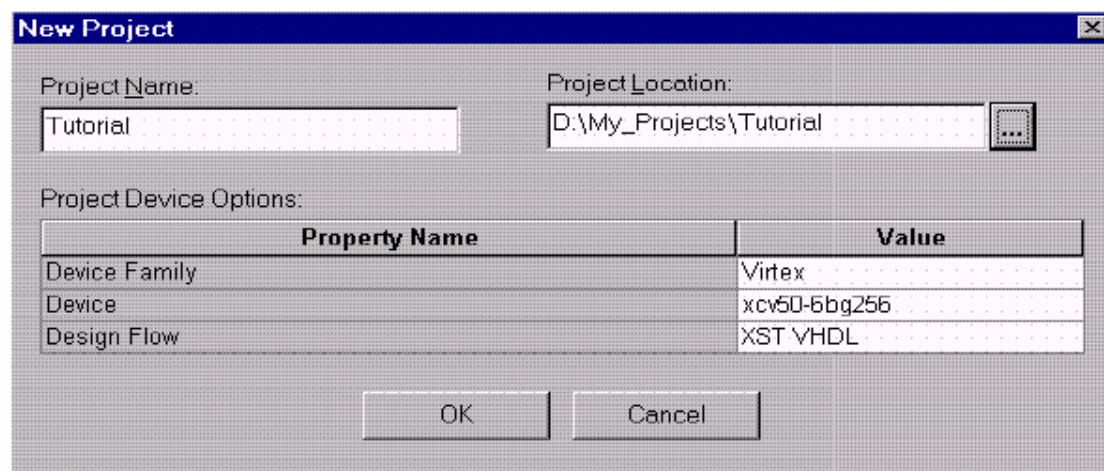


图 1-1 新工程项对话框 (New Project Dialog Box)

5. 点击 OK。

ISE 将会在工程项导航器 (Project Navigator) 中创建和显示你的新工程项。

创建一个计数器模块

按照以下步骤创建一个计数器 VHDL 模块：

1. 选择 Project -> New Source；
2. 选择 VHDL 模块 (VHDL Module) 作为源程序类型；
3. 在文件名中键入 ‘counter’；

4. 点击 Next;
5. 点击 Next;
6. 点击 Finish ,完成这个新源程序的模板。新源程序模板 counter.vhd 将会显示在 HDL 编辑窗口中, 它包括计数器的 Library, Use, Entity, Architecture 等语句的描述。

利用计数器模板修改你的计数器模块

要完成这个计数器模块, 你将要用到 ISE 的语言模板 (ISE Language Template) 工具, 选择其中的计数器 VHDL 描述来完成 counter.vhd 的设计。

1. 选择 Edit -> Language Templates 来打开语言模板, 或者通过点击工具栏中最右端的灯泡按钮来打开语言模板;



2. 在语言模板 (Language Templates) 窗口中, 通过点击 “+” 符号来展开 VHDL 下的综合模板 (Synthesis Templates);

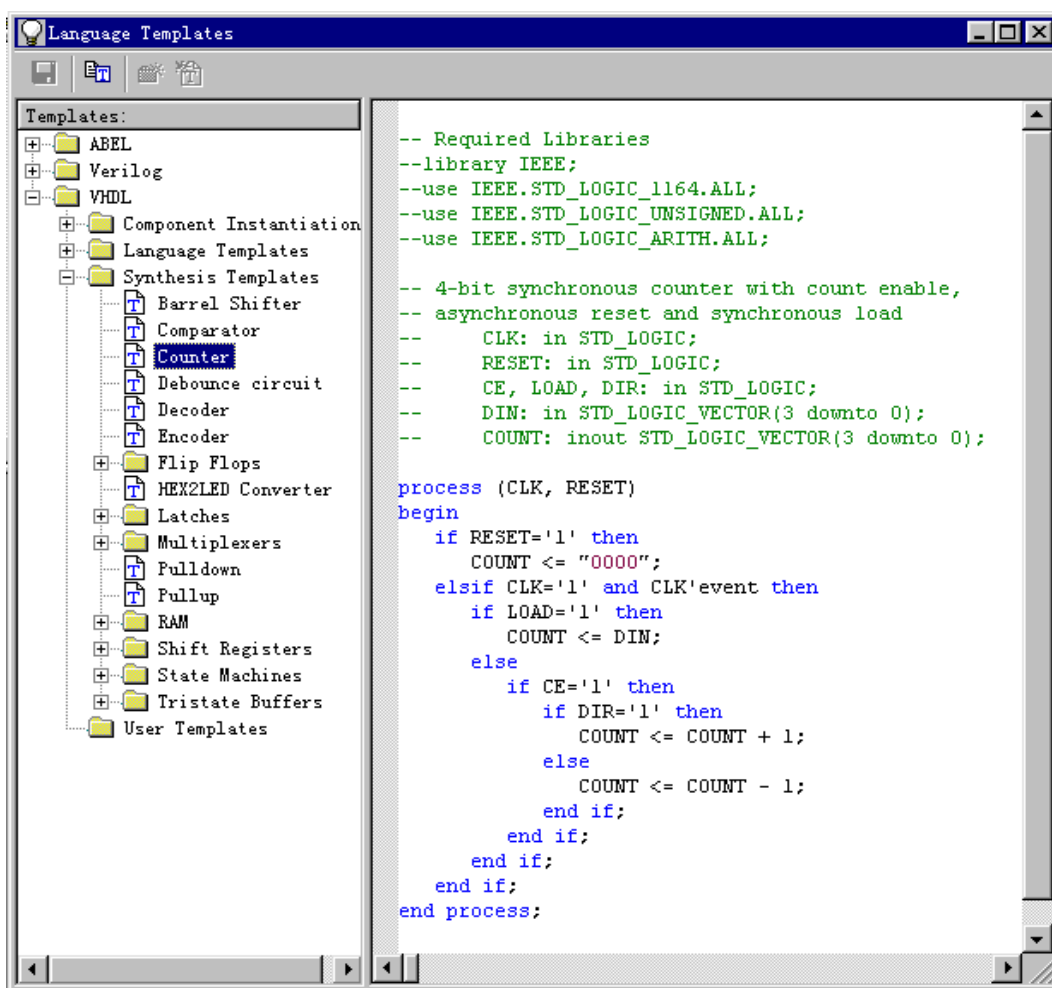
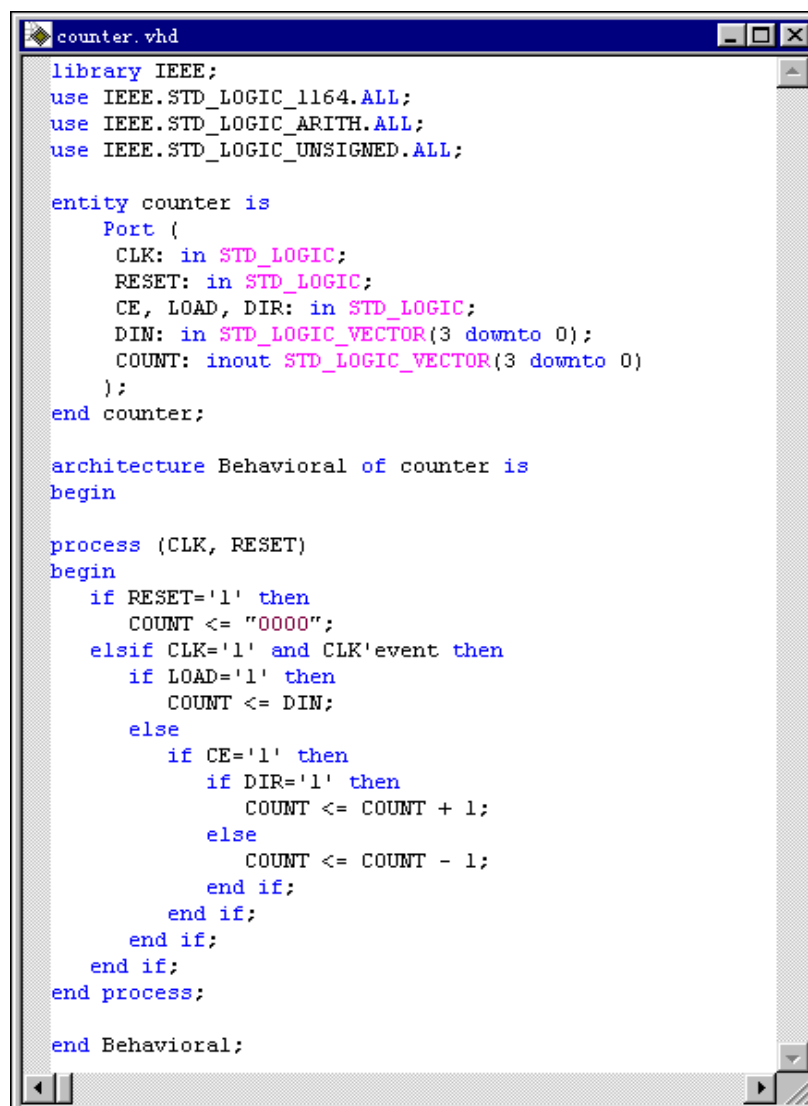


图 1-2 计数器语言模板 (Counter Language Template)

3. 从 VHDL 综合模板中选择计数器模板 (Counter Template), 并把它拖动或粘贴到源程序 counter.vhd 的 begin 和 end 之间;
 4. 关闭语言模板 (Language Templates) 窗口;
 5. 将带有注释符号 (--) 的计数器端口定义的语句剪切并粘贴到计数器的实体 (entity) 描述中去。计数器端口定义语句如下:
-- CLK: in STD_LOGIC;
-- RESET: in STD_LOGIC;
-- CE, LOAD, DIR: in STD_LOGIC;
-- DIN: in STD_LOGIC_VECTOR(3 downto 0);
-- COUNT: inout STD_LOGIC_VECTOR(3 downto 0);
 6. 去掉 counter.vhd 源程序中端口定义语句前的注释符号 (--);
 7. 去掉 COUNT 端口定义语句后的分号 (;), 如下所示:
COUNT: inout STD_LOGIC_VECTOR(3 downto 0)
 8. 选择 File -> Save, 保存 counter.vhd 源程序。
- 此时, 你的 counter.vhd 源程序应与 图 1-3 中的 VHDL 描述一样。



```
counter.vhd
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity counter is
  Port (
    CLK: in STD_LOGIC;
    RESET: in STD_LOGIC;
    CE, LOAD, DIR: in STD_LOGIC;
    DIN: in STD_LOGIC_VECTOR(3 downto 0);
    COUNT: inout STD_LOGIC_VECTOR(3 downto 0)
  );
end counter;

architecture Behavioral of counter is
begin

process (CLK, RESET)
begin
  if RESET='1' then
    COUNT <= "0000";
  elsif CLK='1' and CLK'event then
    if LOAD='1' then
      COUNT <= DIN;
    else
      if CE='1' then
        if DIR='1' then
          COUNT <= COUNT + 1;
        else
          COUNT <= COUNT - 1;
        end if;
      end if;
    end if;
  end if;
end process;

end Behavioral;
```

图 1-3 修改后的计数器模块 (Modified Counter Module)

仿真行为模型（功能仿真）

在本节中，你将会创建一个 testbench 波形，用于定义计数器模块所应有的功能。这个 testbench 波形将被用于与 Modelsim 仿真器连接，用来验证该计数器是否达到设计要求的功能和时延需求。

创建一个 Testbench 波形源文件

首先，在 Project Navigator 中创建一个 testbench 波形源文件，该文件将在 HDL Bencher 中进行修改。

1. 在工程项窗口（Project Window）的源文件中选中计数器(counter.vhd)；
2. 选择 Project -> New Source；
3. 在新的对话框中，选择新文件类型为 Test Bench Waveform；
4. 键入文件名为“counter_tbw”；
5. 点击 Next ；
注：在其他工程项中，你可以将你的 testbench 波形与其他源文件关联。
6. 点击 Next ；
7. 点击 Finish ；

此时，HDL Bencher 程序自动启动并等候你输入所需的时序需求。

你现在可以指定仿真所需的时间参数。时钟高电平时间和时钟低电平时间一起定义了设计操作必须达到的时钟周期，输入建立时间定义了输入在什么时候必须有效，输出有效延时定义了有效时钟沿到达后多久必须输出有效数据。

在本教程中，不需要改变任何默认的时间约束。默认的初始化时间设置如下：

- 时钟高电平时间（Clock high time）：50 ns
- 时钟低电平时间（Clock low time）：50 ns
- 输入建立时间（Input setup time）：10 ns
- 输出有效时间（Output valid delay）：10 ns

8. 点击 OK，接受默认的时间设定。

你的 testbench 波形显示如图 1-4 所示：

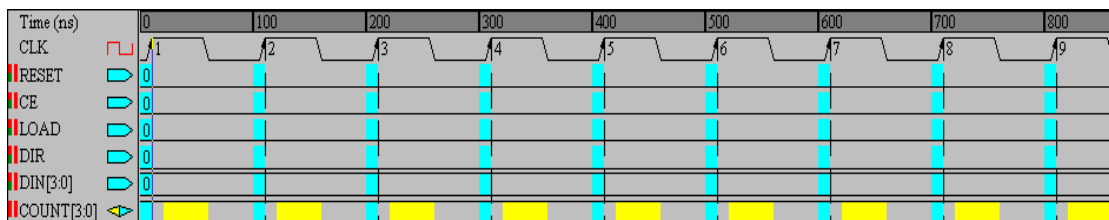


图 1-4 在 HDL Bencher 中的 Testbench 波形

初始化计数器输入

在 HDL Bencher 的波形中，初始化计数器输入如下：

（注：在每个单元的蓝色区域输入激励。）

1. 在 CLK 第 1 周期下点击 RESET 单元直到该单元变为高；
2. 在 CLK 第 2 周期下点击 RESET 单元直到该单元变为低；
3. 在 CLK 第 3 周期下点击 CE 单元直到该单元变为高；
4. 在 CLK 第 2 周期下点击 DIR 单元直到该单元变为高；

你的 testbench 波形现在看起来应该如图 1-5 所示。

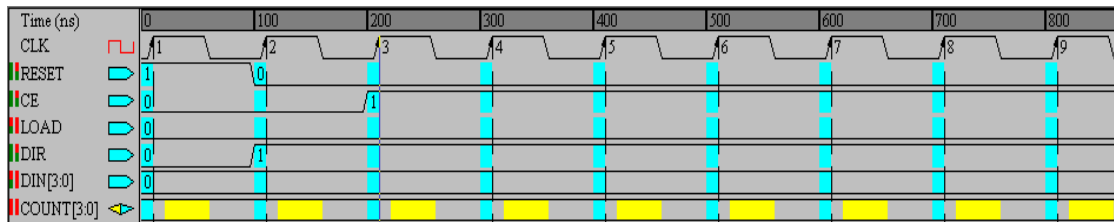


图 1-5 HDL Bencher 激励及响应输入

5. 将你的 testbench 文件存盘。选择 File -> Save Waveform，或点击工具栏的存盘图标。接下来，HDL Bencher 会提示你设置你希望仿真的时钟周期数；



6. 在“End the testbench __ cycles after the last input assignment”对话框中输入8，默认值为1；
7. 点击 OK；
8. 退出 HDL Bencher。

新的 testbench 波形源文件 (counter_tbw.tbw) 自动加入到该工程中。

生成预期的输出响应

现在，你可以在已输入初始化激励的基础上，生成时钟模块预期的输出。

1. 在 Sources in Project 窗口中，选中 counter_tbw.tbw 文件；
2. 在 Processes for Current Source 窗口中，点击 ModelSim Simulator 旁边的“+”符号展开 ModelSim 仿真器的层次结构；
3. 双击 Generate Expected Simulation Results；
此步骤采用定义好的输入运行一个后台仿真，产生输出值加入到 testbench 波形中。

你的 testbench 波形现在看起来应该如图 1-6 所示。

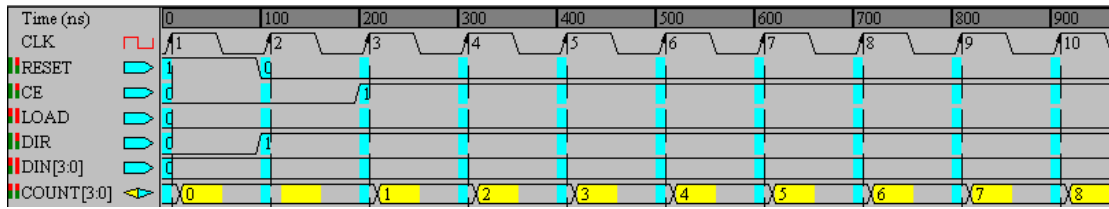


图 1-6 生成仿真结果

4. 直接退出 HDL Benchner 而不选择波形存盘。

使用 ModelSim 进行仿真

根据 HDL Benchner 中产生的预期结果，你现在可以用 ModelSim 进行仿真。在这部分教程中，你将运行行为仿真（亦即功能仿真）和布局布线后的仿真（时序仿真）。

行为仿真

我们进行行为仿真以验证计数器模块的功能。

1. 在 Sources in Project 窗口中选择 counter_tbw.tbw 文件；
2. 在 Processes for Current Source 窗口中，点击 “+” 符号展开 ModelSim 仿真器的层次结构，找到并双击 Simulate Behavioral VHDL Model；
此时，ModelSim 仿真器自动启动。
3. 对于第一次运行 ModelSim 的用户，会显示一个对话框，需要在其中做以下处理：

- 选中 Do not show this dialog again 选项
- 点击 Run ModelSim

此对话框在你重新安装或重新配置 ModelSim 之前将不再显示。

你的仿真结果现在显示在 ModelSim 的波形窗口(wave window)。

注：ISE 通过自动创建及运行一个仿真宏文件(.fdo文件)来自动处理仿真过程。虽然用户无法看到，但在本教程中，counter_tbw.fdo 文件实现了以下操作：

- 创建设计库；
 - 编译设计源文件和 testbench 源文件；
 - 调用仿真器；
 - 打开所有的观察窗口；
 - 将所有信号加到 wave 窗口；
 - 将所有信号加到 list 窗口；
 - 按照要求的运行时间属性进行仿真(默认为1000ns)。
4. 点击 Zoom -> Zoom Ful ，或在工具条点击 Zoom Full 图标；



5. 点击 Zoom -> Zoom in ，或在工具条点击 Zoom in 图标；



6. 拖动波形窗口下端的滚动条至窗口的最左端，仿真输出波形应如图 1-7 所示。

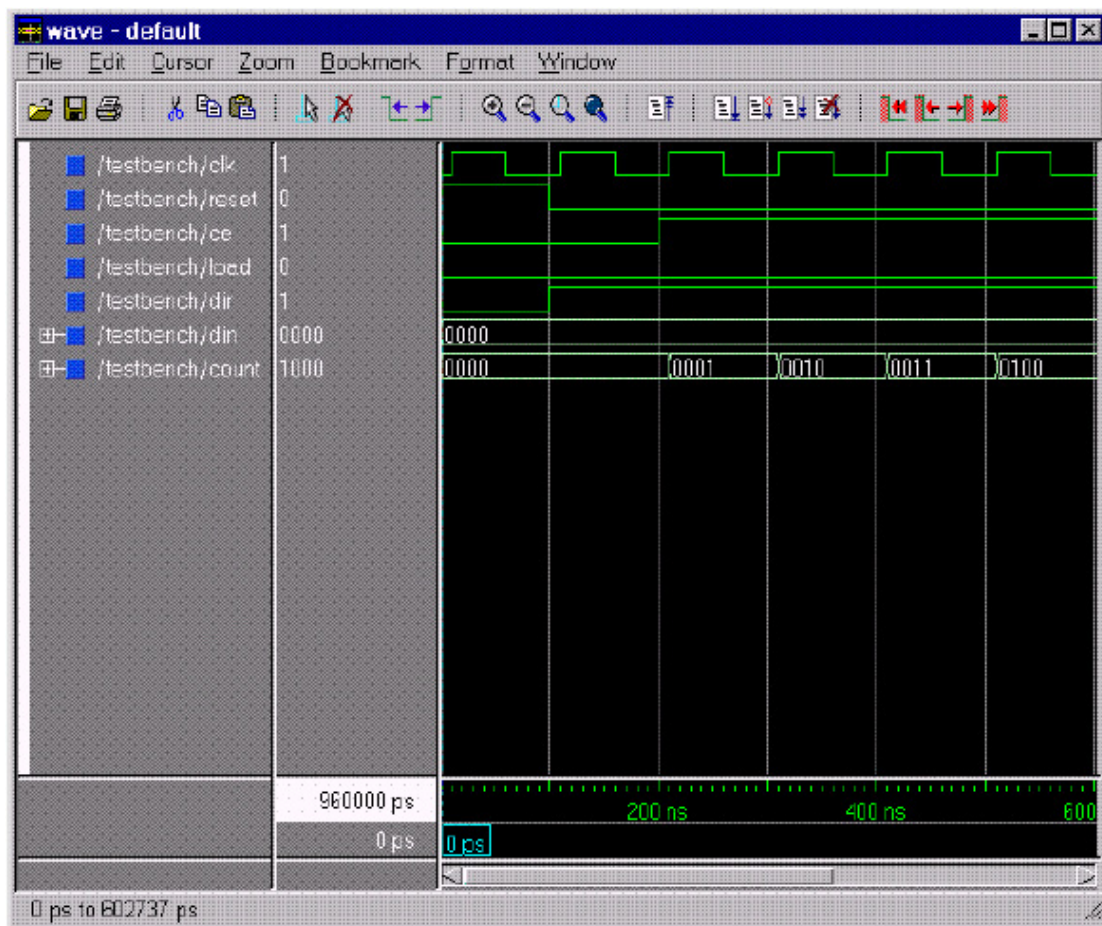


图 1-7 行为仿真波形

7. 关闭 ModelSim 主窗口，退出 ModelSim。

布局布线后的仿真

布局布线后的仿真包括了目标器件的时延信息。对 counter 模块的后仿真操作如下：

1. 在 Sources in Project 窗口中选择 counter_tbw.tbw;
2. 在 Processes for Current Source 窗口，点击 “+” 符号 展开 ModelSim 仿真器的层次结构，找到并双击 Simulate Post-Place & Route VHDL Model;
注：此步骤会启动设计实现来进行布局布线及启动ModelSim。
3. 点击 Zoom -> Zoom Full ，或在工具条中点击 Zoom Full 图标。



4. 点击 Zoom -> Zoom In ，或在工具条中点击 Zoom in 图标。



5. 拖动波形窗口下端的滚动条至波形窗口的最左端。仿真输出波形应该如图 1-8

所示。

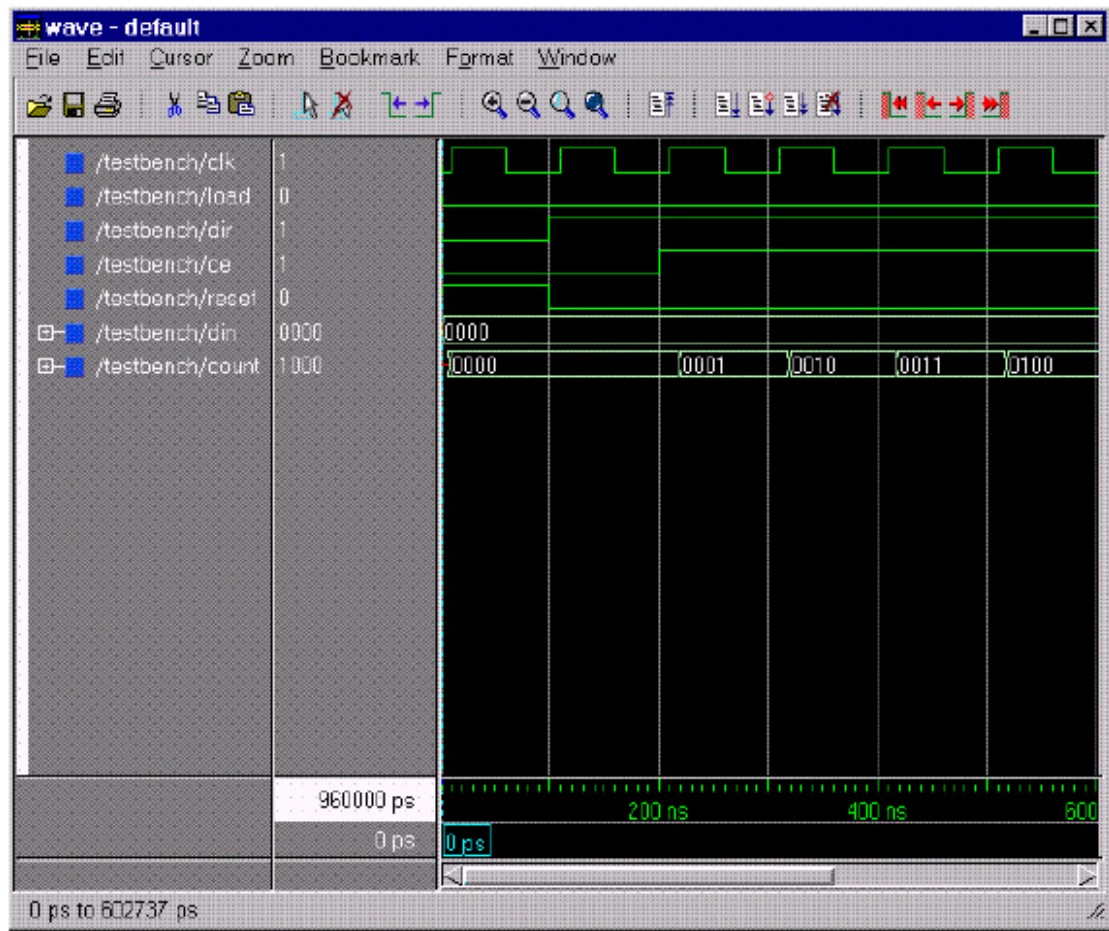


图 1-8 布局布线后的仿真波形

6. 关闭 ModelSim 主窗口，退出ModelSim。

设计输入 (顶层为原理图) (Top-Level Schematic)

本节以一个原理图设计为例, 说明了如何创建一个包含有例化的 VHDL 计数器模块的顶层原理图, 描述了如何将若干个模块连接成一个原理图, 如何给单根连线命名, 如何给总线命名以及如何添加输入/输出管脚标记。

创建 VHDL 模块 (VHDL Module), 生成一个原理图符号 (Schematic Symbol)

1. 在工程项导航器 (Project Navigator) 菜单中, 点击 Project -> Add Source 来添加一个 VHDL 模块 -- counter.vhd。(在选择源类型(Choose Source Type)弹出窗口中选择 VHDL Module);
2. 在 Sources in Project 窗口中, 选中计数器模块 counter.vhd;
3. 在 Processes for Current Source 窗口中, 点击设计输入实用程序 (Design Entry Utilities) 之前的 “+” 符号, 然后双击创建原理图符号 (Create Schematic Symbol)。

经过以上步骤, 名称为 “counter” 的图形化元件被放入到工程项库中。

创建一个新的顶层原理图 (Top-Level Schematic)

1. 在工程项导航器 (Project Navigator) 菜单中, 选择 Project -> New Source;
2. 选择原理图 (Schematic) 为源类型;
3. 输入原理图名为 “top”;
4. 先点击 “Next”, 再点击 “Finish”。

此时, 原理图编辑器 (ECS) 自动启动并在其原理图窗口中打开一张空图。

例化 VHDL 模块 (Instantiating VHDL Module)

在原理图编辑器 (ECS) 中, 例化两个 VHDL 计数器模块 -- counter

1. 在菜单中选择 Add -> Symbol, 或者在工具栏中点击 (Add Symbol) 图标;



2. 从元件符号列表 (在屏幕右侧) 中, 选择计数器 counter;
注意, 不要在类别 (Categories) 窗口中作任何选择。
3. 点击左键可将计数器 counter 放置在光标所在的位置;
请在原理图中放置两个计数器 counter。现在, 原理图应该如图 1-9 所示。
4. 按 ESC 键退出添加符号 (Add Symbol) 模式。

注: 可以通过放大缩小功能 (View -> Zoom -> In / out / To Box / Full View) 和拖动滚动条来调整视图大小及位置。

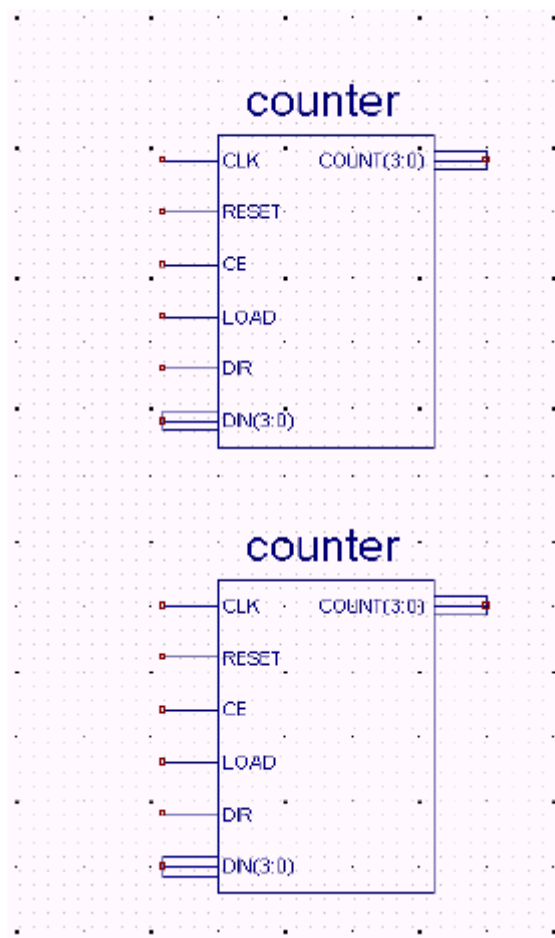


图 1-9 例化 VHDL 模块

原理图中连线 (Wiring the Schematic)

在连接元件符号时，一些连线端可以悬空，而另一些连线端需和元件符号相连。

1. 首先激活划线功能，通过在菜单中选择 Add -> Wire，或者在工具栏中点击 (Add Wire) 图标；



2. 添加一根悬空线和延展连线
 - a) 在计数器模块的某一管脚单击鼠标；
 - b) 将连线拉伸到需要的长度；
 - c) 在连线端点处双击鼠标；

按照如图 1-10 所示，给计数器模块的每一管脚添加连线。
3. 添加两个元件符号之间的连线
 - a) 在一个计数器模块的管脚处单击鼠标；
 - b) 在另一个计数器模块的对应管脚处双击鼠标，结果如图 1-10 所示。

当完成连线后，按 ESC 键退出添加连线 (Add Wire) 模式。

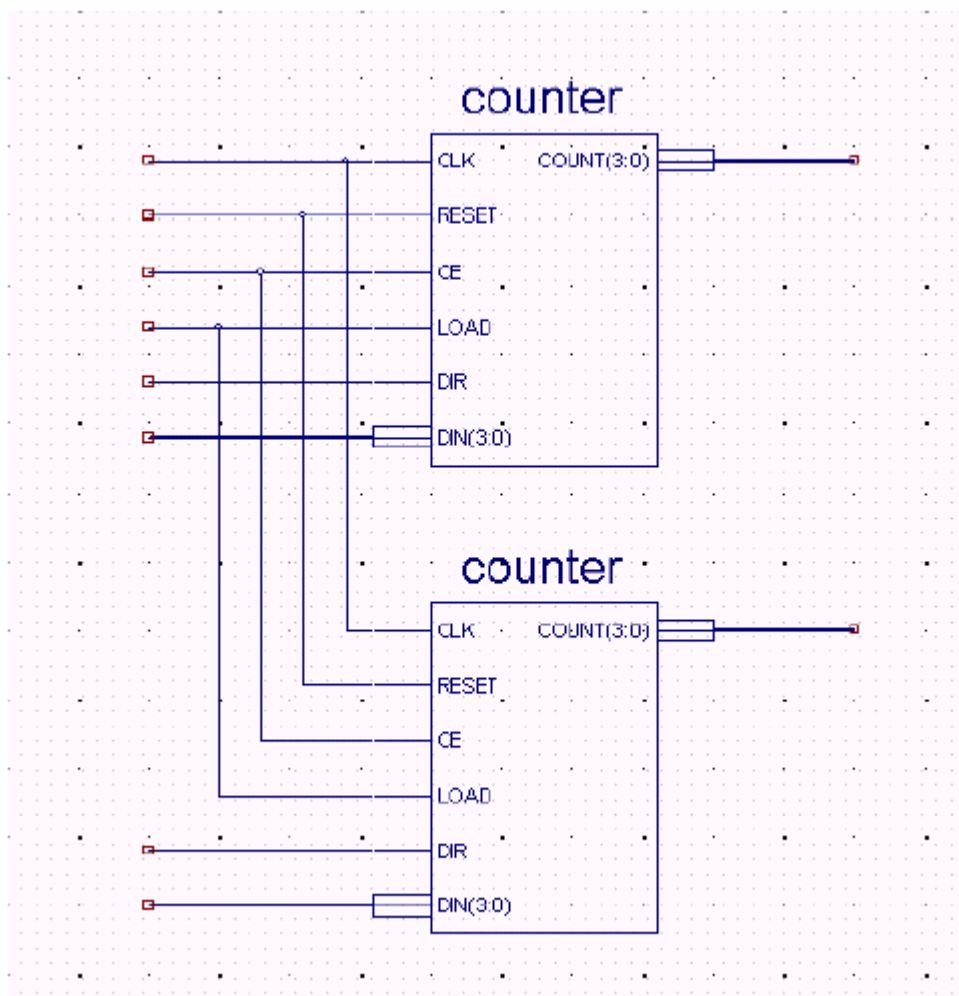


图 1-10 为模块相应管脚连线

为连线添加网络名 (Add Net Name to wires)

画好连线后，你可为相应的连线添加网络名称。

1. 在菜单中选择 Add -> Net Name，或者在工具栏中点击 (Add Net Name) 图标，



2. 创建网络名并放置到连线上的步骤如下：
 - a) 在工具栏右边的文本对话框中输入网络名 clock，如图 1-11 所示；
注：使用网络命名 (Name Branch) 和保持网络名 (Keep Name) 的默认值。



图 1-11 添加网络名

- b) 将鼠标移动到悬空连线的端点处（此时光标处显示出网络名）；
- c) 点击鼠标左键；

注：按照上面步骤添加其余 5 个网络名：reset, ce, load, dir1 和 dir2;

- 按 ESC 键退出添加网络名(Add Net Name)模式。

添加完 6 个网络名后，原理图应该如图 1-12 所示。

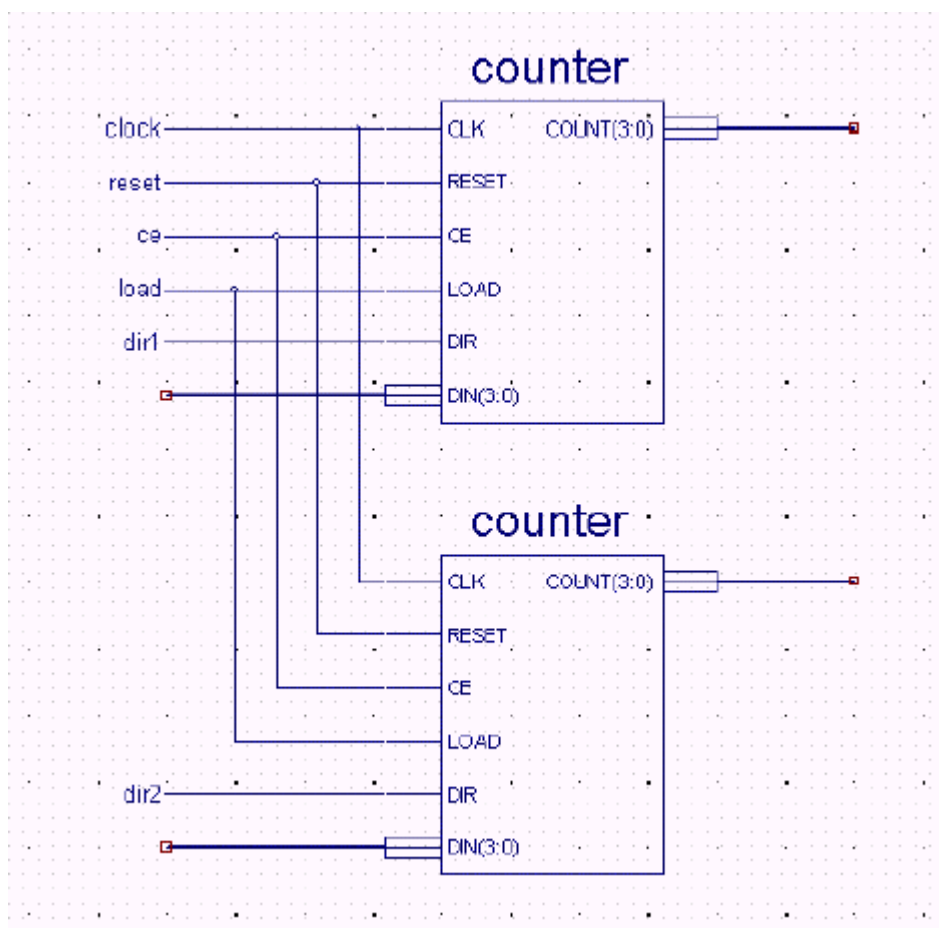


图 1-12 为连线添加完网络名后的原理图

为总线添加网络名

按照同样的步骤，为计数器模块的 din 和 count 管脚添加总线网络名并给总线宽度赋值。

- 在菜单中选择 Add -> Net Name，或者在工具栏中点击 (Add Net Name) 图标；



- 创建总线网络名并放置到总线上的步骤如下：
 - 在工具栏右边的文本对话框中输入总线名和宽度，如：din1(3:0)；
注：使用网络命名 (Name Branch) 和保持网络名 (Keep Name) 的默认值。
 - 将鼠标移动到悬空总线的端点（此时光标处显示出总线名及宽度）；
 - 点击鼠标左键；
注：按照上面步骤添加其余 3 个总线名：count1(3:0)，count2(3:0) 和

din2(3:0);

3. 按 ESC 键退出添加网络名 (Add Net Name) 模式。

添加完总线名后, 原理图应该如图 1-13 所示。

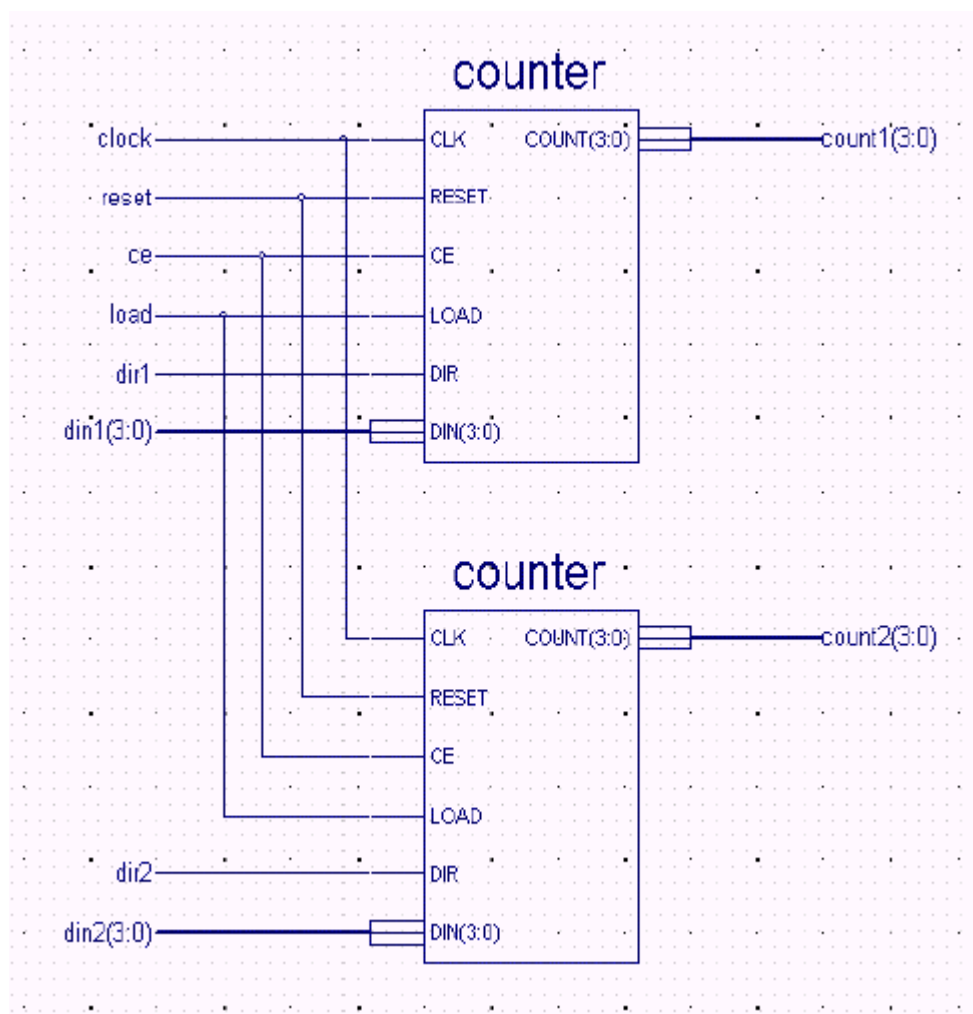


图 1-13 为总线添加完网络名后的原理图

添加输入/输出管脚标记 (Adding I/O Markers)

根据各个输入/输出信号的方向, 添加相应的标记。在本教程中, 我们要添加输入标记和双向信号标记, 最后结果应如图 1-14 所示。

1. 在菜单中选择 Add → I/O Marker, 或在工具栏中点击 (Add I/O Marker) 图标;



2. 按如下步骤为 clock, reset, ce, load, dir1 和 dir2 添加输入标记, 同时为总线 din1(3:0) 和 din2(3:0) 添加输入标记:

- 在工具栏右边的参数单选框中, 选择输入 (Input);
- 将鼠标移动到输入信号线的端点 (此时光标处显示出输入标记的图形);
- 点击鼠标左键 (输入标记会将网络名或总线名包含在标记图形的内部);

注：一定要在信号线的端点处单击鼠标。如果添加输入/输出标记不成功，系统会弹出错误对话框。

3. 按如下步骤为 count 总线添加双向信号标记：
 - a) 在工具栏右边的参数单选框中，选择双向(Bidirectional)；
 - b) 将鼠标移动到输出信号线的端点(此时光标处显示出双向信号标记的图形)；
 - c) 单击鼠标左键；

完成的原理图如图 1-14 所示。

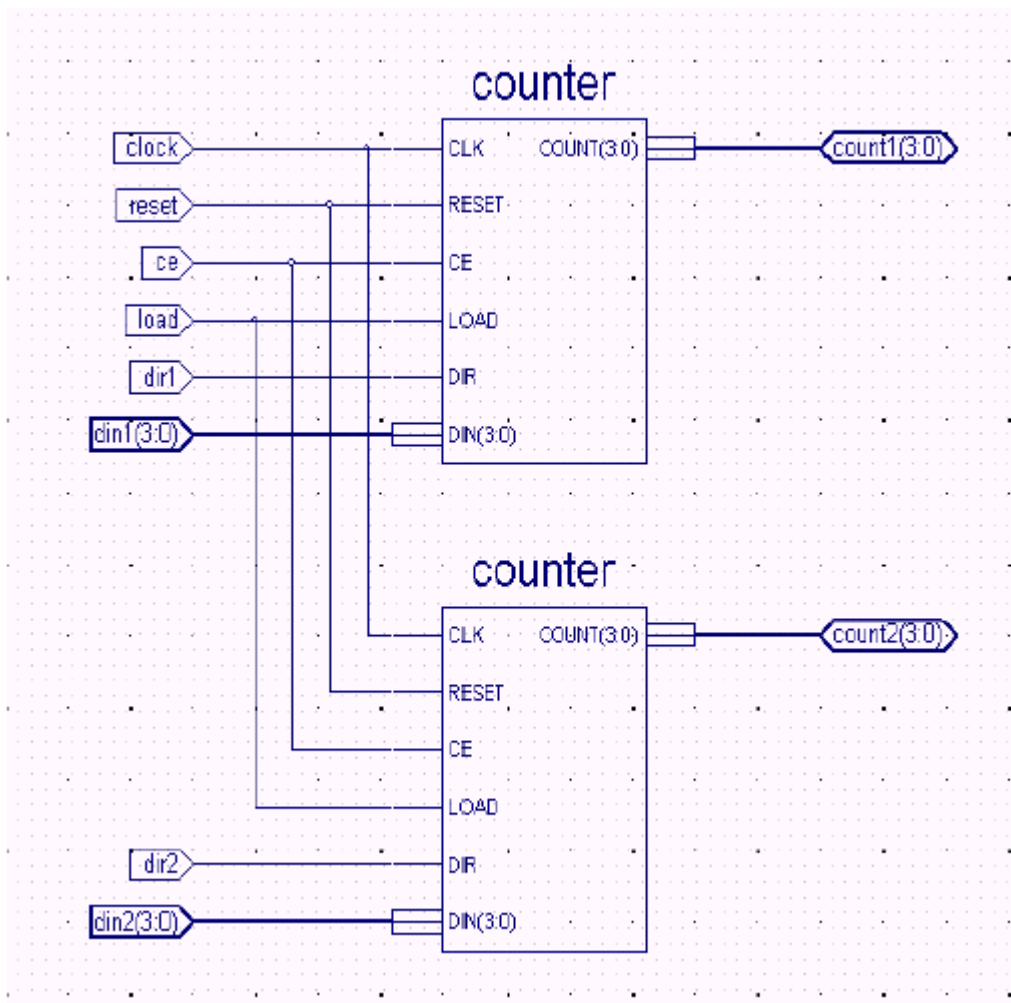


图 1-14 完成的原理图

4. 在菜单中选择 File -> Save, 保存原理图；
5. 退出原理图编辑器 (ECS)。

设计实现

在本教程中，设计实现部分涵盖了两项任务：在工程项导航器(Project Navigator)中运行设计实现（Implement Design）进程，以及在资源分配器（Floorplanner）工具中查看设计在布局布线后的结果。

运行设计实现

首先，运行所有与这个计数器设计相关联的进程（从综合到布局布线）。在原理图文件上启动设计实现（Implement Design）以完成操作：

1. 在 Sources in Process 窗口中选择 top (top.sch)；
2. 在 Processes for Current Source 窗口中双击设计实现（Implement Design）。

这样就运行了所有相关的进程。

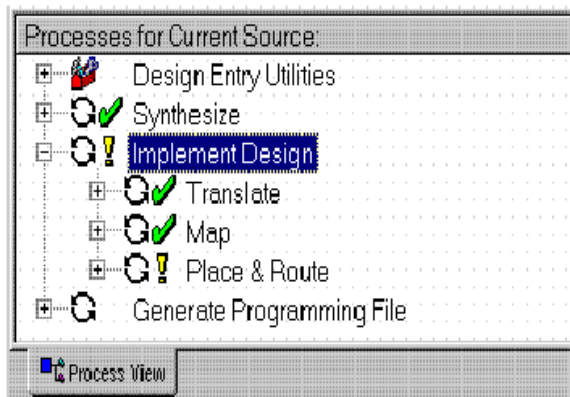


图 1-15 设计实现过程

在 Process for Current Source 窗口中，打对勾的标记指示进程已经成功地运行；感叹号标记表示进程已经运行，但是包含有系统给出的警告，有关警告的更多信息可以从副本（Transcript）窗口中获取。

在资源分配器（Floorplanner）中查看设计布局

现在，你可以在资源分配器（Floorplanner）中查看完成后的设计。

1. 在 Source in Process 窗口中选择 top (top.sch)；
2. 在 Process for Current Source 窗口中点击 Implement Design 左边的“+”符号，展开层次后，再点击 Place & Route 左边的“+”符号；
3. 双击 View/Edit Placed Design (Floorplanner)，资源分配器（Floorplanner）将自动运行并显示这个工程项中设计的布局情况。

通过显示和放大输入/输出信号，可以更有针对性地查看设计实现后的结果。

1. 在 top.fnf Design Hierarchy 窗口中（View -> Hierarchy）选择最顶层，—— ‘top (22 IOBs, 13 FGs, 8 CYs, 8 DFFs, 1 BUFG) ’，在布局（Placement）窗口显示这些信号；

注：也可以选择布局（Placement）窗口的设计区域画一个矩形框来显示信号。

2. 在菜单中选择 View -> Zoom -> To Selected，或者在资源分配器的工具条中点击 Zoom to Selected 图标（工具条中右边的最后一个图标）；



3. 将光标放置在每一个引脚上，在资源分配器左下角可以读出该引脚的名称，便于校验所有的输入输出。

注：在 top.fnf Design Hierarchy 窗口的列表中单独选择一个信号，可以在布局窗口中独立地观察这个信号。

在资源分配器（Floorplanner）中看见的布局情况如图 1-16 所示。

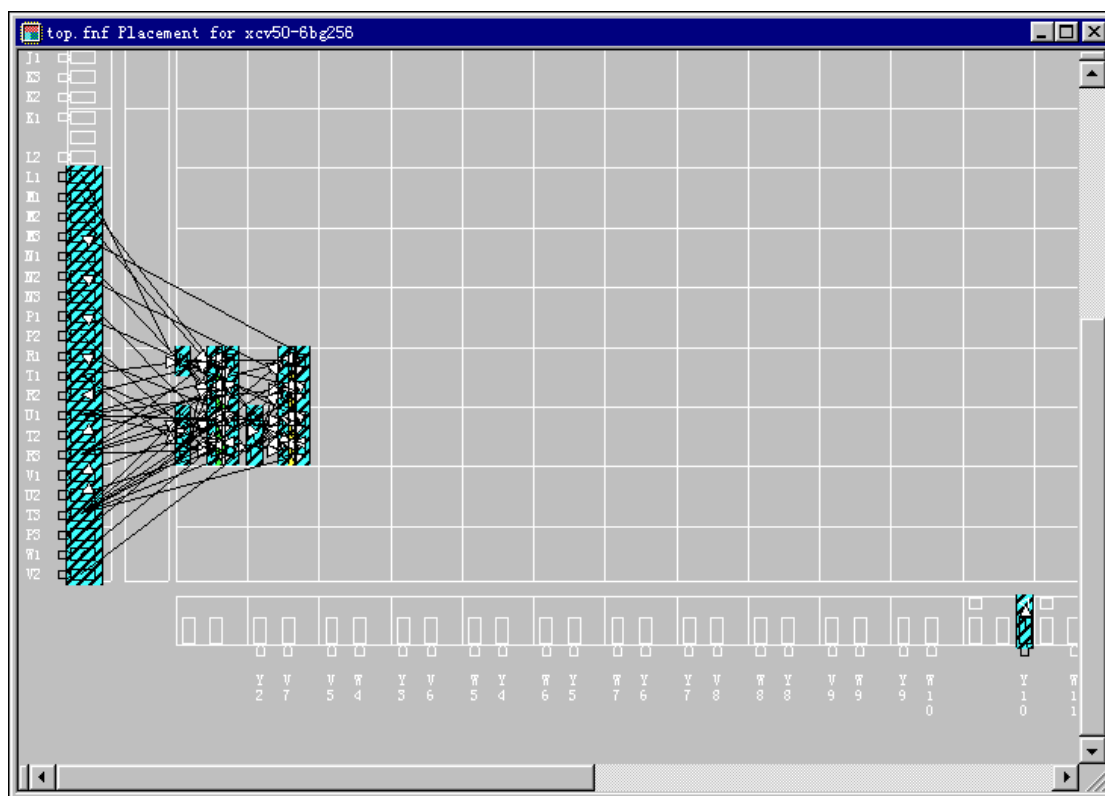


图 1-16 资源分配器（Floorplanner）中输入输出信号的连接情况

当完成了对设计实现布局的观察后，点击 File -> Save，保存资源分配器的设计视图并退出资源分配器。

对顶层文件进行仿真（时序仿真）

接下来，对在前面部分所创建的顶层设计进行时序仿真。首先，使用 HDL Bencher 为顶层设计创建一个测试矢量（testbench）波形，然后用 ModelSim 对顶层设计进行仿真。

建立一个 testbench 波形源文件

创建一个可以在 HDL Bencher 中修改的测试矢量（testbench）波形。

1. 在 Project Navigator 的 Source in Project 窗口中选择 top (top.sch);
2. 选择 Project -> New Source;
3. 在 New 对话框中选择 Test Bench Waveform 作为源文件类型;
4. 键入文件名 “top_tbw”;
5. 选择 Next;
6. 确认 top 是关联的源文件，然后选择 Next;
7. 选择 Finish;

HDL Bencher 自动启动。

8. 点击 OK，让测试矢量（testbench）波形使用默认的时序约束。

初始化计数器输入

在 HDL Bencher 的波形窗口中，按下面的步骤初始化计数器的输入，与图 1-17 进行比较来检查你的输入。

注：在每一个信号单元的蓝色区域加入输入激励。

1. 在 clock 周期 3 下面点击 ce 单元直至该信号被置高;
2. 在 clock 周期 2 下面点击 dir1 单元直至该信号被置高;
3. 在 clock 周期 1 下面点击 dir2 单元直至该信号被置高;
4. 在 clock 周期 2 下面点击 dir2 单元直至该信号被置低;
5. 在 clock 周期 1 下面点击 reset 单元直至该信号被置高;
6. 在 clock 周期 2 下面点击 reset 单元直至该信号被置低。

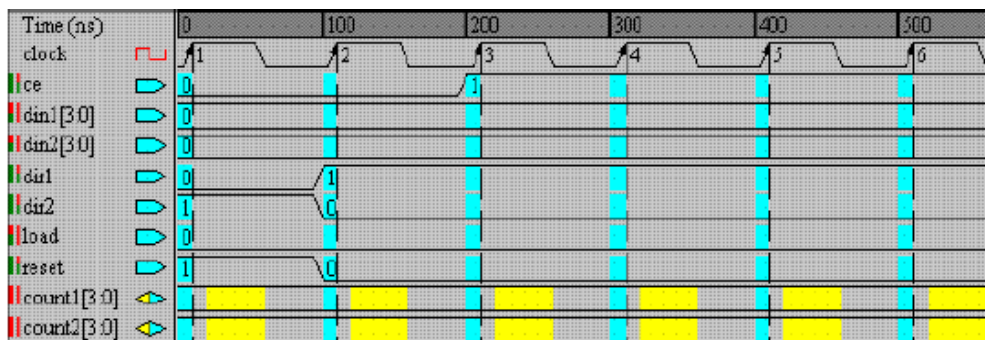


图 1-17 HDL Bencher 激励与响应的引入

生成预期的输出响应

生成预期的响应，需在 HDL Bencher 波形中的黄色区域做出以下的操作：

1. 在 clock 周期 2 下面点击 count1(3:0) 的黄色单元；
2. 点击 Pattern 按钮启动模式向导 (Pattern Wizard)；
3. 在 Pattern Wizard 的对话框中设置参数，使得期望的计数输出值为 0 到 7，如下所示：

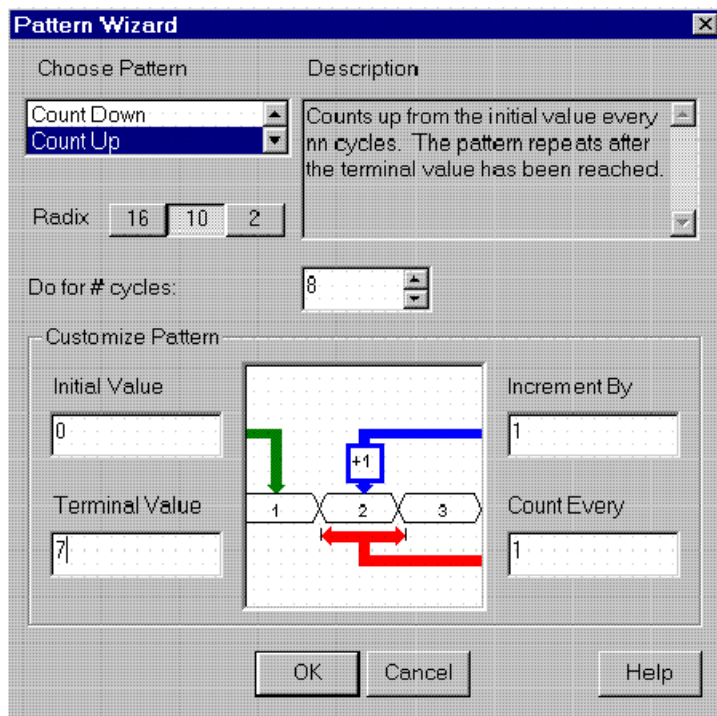


图 1-18 Pattern Wizard 设置

4. 在 Pattern Wizard 的对话框中点击 OK；
5. 在 clock 周期 2 下面点击 count2(3:0) 的黄色单元并键入 0；
6. 在 clock 周期 3 下面点击 count2(3:0) 的黄色单元；
7. 点击 Pattern 按钮启动模式向导 (Pattern Wizard)；
8. 在 Pattern Wizard 的对话框中设置参数，使得期望的计数输出值为 15 到 9，如图 1-19 所示。
9. 在 Pattern Wizard 的对话框中点击 OK，测试矢量波形如图 1-20 所示。
10. 在菜单中选择 File -> Save，或者在工具栏中点击 Save Waveform 图标保存测试矢量波形；



11. 退出 HDL Bencher。

注：在 Project Navigator 的 Source in Project 窗口中，新的测试矢量 (testbench) 波形文件是顶层文件 (top. sch) 的一个子文件。

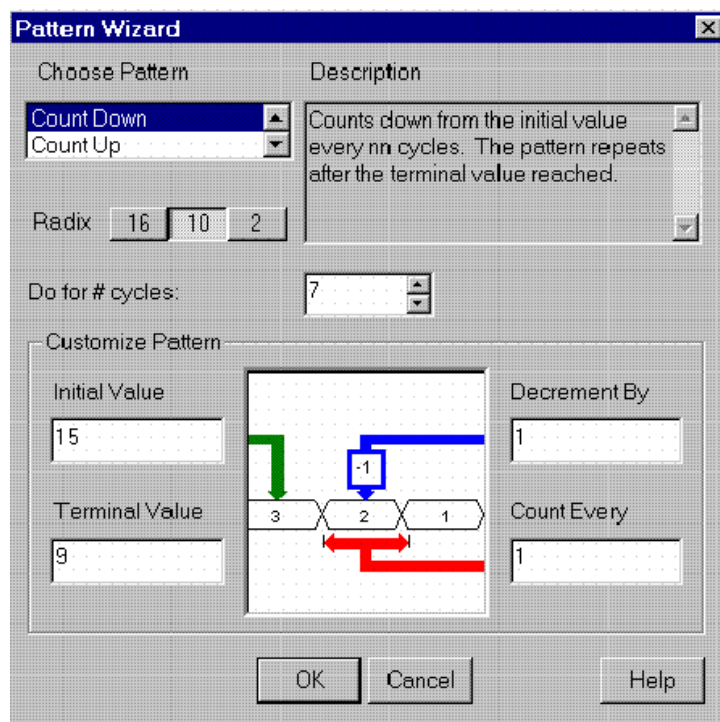


图 1-19 Pattern Wizard 设置

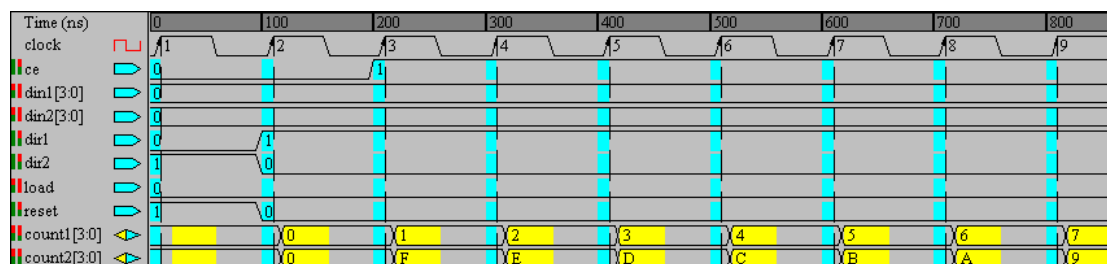


图 1-20 测试矢量波形

布局布线后的仿真

对顶层设计执行布局布线后的仿真：

1. 在 Project Navigator 的 Source in Project 窗口中选择 top_tbw.tbw;
2. 在 Processes for Current Source 窗口中，双击 ModelSim Simulator 层次结构中的 Simulate Post-place and Route VHDL Model;
此时，ModelSim 自动启动，同时引入了反标注后的设计文件。
3. 在 wave 窗口中，点击下拉菜单中的 Zoom -> Zoom Full，或者点击工具栏中的 Zoom Full 图标；



4. 点击下拉菜单中的 Zoom -> Zoom In, 或者点击工具栏中的 Zoom In 图标, 并将滚动条拖到波形窗口的最左边;



看到的波形应该如图 1-21 所示。

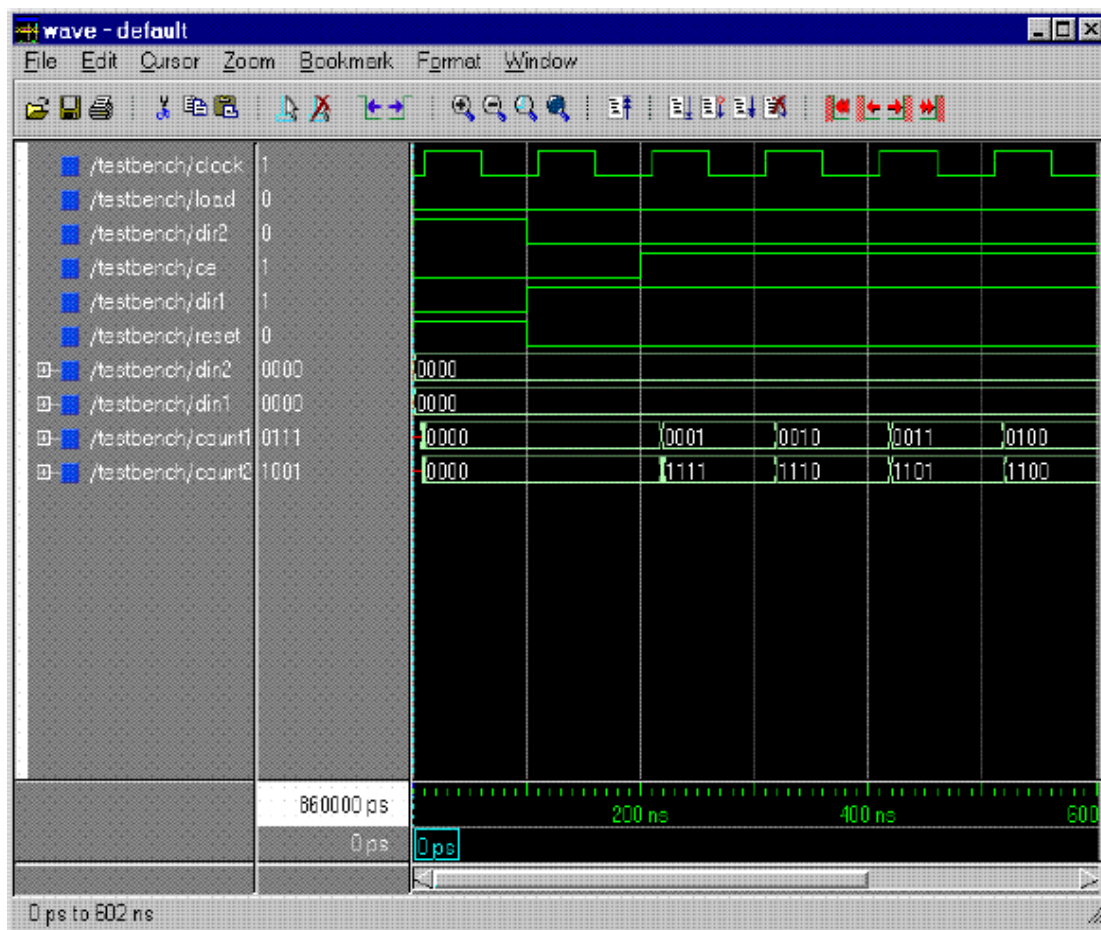


图 1-21 时序仿真波形

5. 时序仿真结果显示有 10ns 的时钟延迟;
6. 分析完你的仿真结果以后, 通过关闭 ModelSim 的主窗口 (main window) 来退出 ModelSim。

附录：EDIF 设计方法

本附录描述了如何在 ISE 环境中使用 EDIF 网表文件作为输入来实现一个设计。当用户已有综合好的网表文件时，可以使用此流程进行 FPGA 实现。

本附录包含如下两部分：

1. 设计输入
2. 设计实现

设计输入

首先，在工程项导航器 (Project Navigator) 中创建一个新的工程项；然后，加入 EDIF 源文件；最后，实现该设计。

创建一个新工程项

1. 选择 File -> New Project;
2. 在新工程项对话框 (New Project dialog box) 中，在工程项存放 (Project Location) 区域键入希望的存放位置，或者点击浏览按钮 (browse) 来选择所需要的位置；
3. 在工程项名称 (Project Name) 处键入 “TutorialEDIF”；
注：当你在工程项名称处键入 “TutorialEDIF” 时，一个名称为 TutorialEDIF 的子目录自动创建在你前面希望的位置。例如，你选择的目录是 D:\My_Projects，那么，你创建的工程项将会存在 D:\My_Projects\TutorialEDIF 目录下。
4. 在新工程项 (New Project) 对话框中，使用下拉箭头为每个属性名 (Property Name) 赋值，结果如图 A-1 所示。
 - 器件系列：Virtex
 - 器件：xcv300-6bg352
 - 设计流程：EDIF

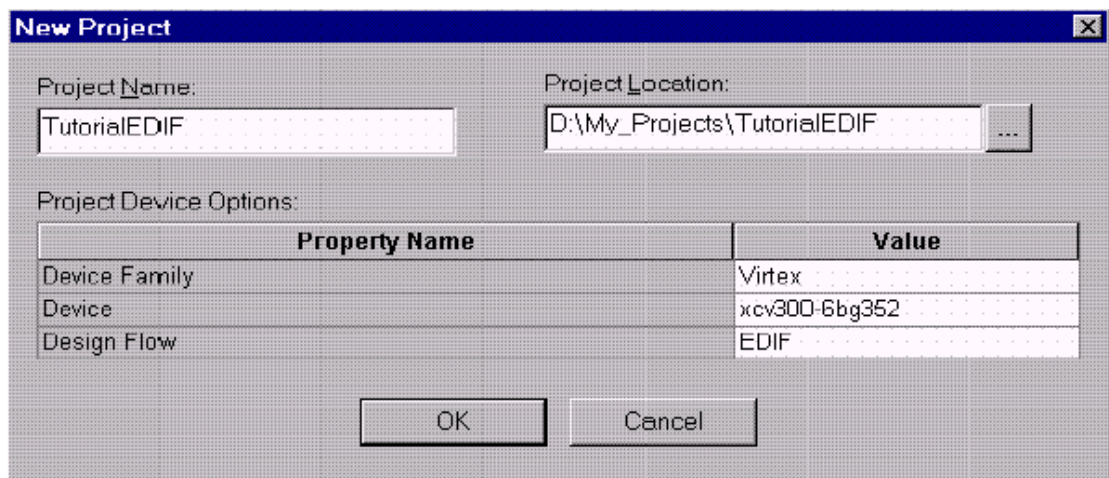


图 A-1 新工程项对话框

5. 点击 OK。

ISE 在工程项导航器中创建了一个子目录和一个新的工程项。

添加 EDIF 源文件

现在，添加一个 EDIF 源文件到这个新的工程项中。本教程中使用的 EDIF 源文件包含在 ISE 安装完的子目录中。按照如下步骤添加源文件：

1. 选择 Project -> Add Source;
2. 使用浏览 (Browse) 功能，在 ISE 的安装目录下找到如下子目录：
/ISEexamples/edif_flow/
3. 选择文件 “mf.edn”;
4. 点击 Open。此时，该文件加到了新工程项中。

注：如果使用 Project -> Add Copy of Source 选项，将会在新工程项子目录中添加一个源文件的备份文件。

设计实现

下一步，实现该设计。同时，使用 FPGA Editor 工具来察看该设计的布局及布线情况。

运行实现设计 (Implement Design)

按照如下步骤实现该设计：

1. 在 Sources in Project 窗口中，选择网表文件 “mf.edn”;
2. 在 Processes for Current Source 窗口中，双击实现设计 (Implement Design)，所有的进程 (从 Translate 到 Place & Route) 将会运行。

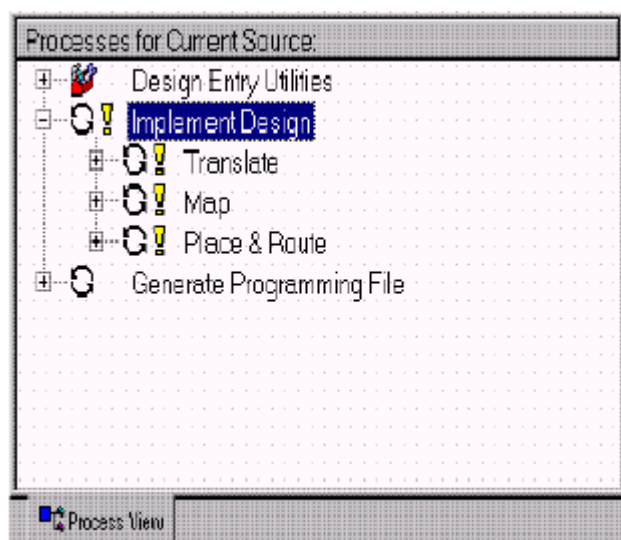


图 A-2 实现设计进程

在 Process for Current Source 窗口中，打对勾的标记指示进程已经成功地运行；感

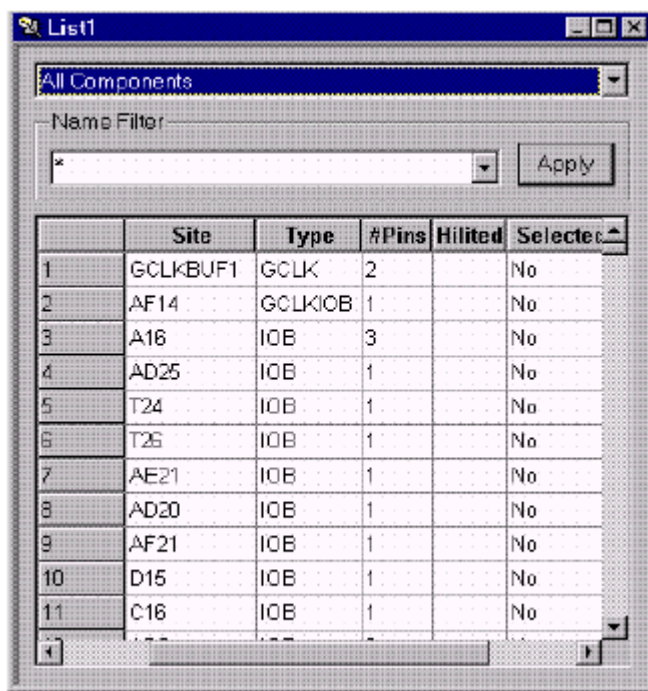
叹号标记表示进程已经运行，但是包含有系统给出的警告，有关警告的更多信息可以从副本 (Transcript) 窗口中获取。

在 FPGA Editor 中查看设计

在 FPGA Editor 中，按照如下步骤查看设计：

1. 当实现完成后，在 Processes for Current Source 窗口中，点击实现设计 (Implement Design) 旁边的 “+” 符号，然后再点击布局布线 (Place & Route) 旁边的 “+” 符号；
2. 双击 View/Edit Routed Design (FPGA Editor)；
布局布线后的设计 -- mf.ncd -- 自动显示在 FPGA Editor 中。
3. 在 List1 窗口中，点击类型 (type) 列将会列出所有按照类型分类的元件。调整表格及列宽到希望的位置，可以清楚地看到各个名称。请确认所有的 26 个输入输出都已在类型列中。

注：时钟输入/输出管脚将会列在 GCLK 类型中，而不是在 IOB 类型中。



| | Site | Type | #Pins | Hilited | Selected |
|----|----------|---------|-------|---------|----------|
| 1 | GCLKBUF1 | GCLK | 2 | | No |
| 2 | AF14 | GCLKIOB | 1 | | No |
| 3 | A16 | IOB | 3 | | No |
| 4 | AD25 | IOB | 1 | | No |
| 5 | T24 | IOB | 1 | | No |
| 6 | T26 | IOB | 1 | | No |
| 7 | AE21 | IOB | 1 | | No |
| 8 | AD20 | IOB | 1 | | No |
| 9 | AF21 | IOB | 1 | | No |
| 10 | D15 | IOB | 1 | | No |
| 11 | C16 | IOB | 1 | | No |

图 A-3 按照类型列出所有元件

完成查看设计后，退出 FPGA Editor。

XILINX 技术资源

Software Manuals (软件手册)

<http://toolbox.xilinx.com/docsan/xilinx4/index.htm>

Tutorials (教程)

<http://support.xilinx.com/support/techsup/tutorials/index.htm>

Data Book (数据手册)

<http://support.xilinx.com/partinfo/databook.htm>

Application Notes (应用指南)

<http://support.xilinx.com/apps/appsweb.htm>

Answers Database (问题解答数据库)

<http://support.xilinx.com/support/searchtd.htm>

Technical Tips (技术提示及设计技巧)

<http://support.xilinx.com/support/techsup/journals/index.htm>

Xcell Journals Quarterly (XILINX 季刊杂志)

<http://support.xilinx.com/xcell/xcell.htm>

致 谢

本教程为《ISE 4 Tutorial》的中文译本。在本教程的编写过程中，香港科汇（亚太）有限公司盈丰分部的 XILINX 现场应用工程师涂平、左峰、魏洁、温广翔和王庆海参加了翻译工作，张全胜和王庆海对全文进行了校对及整理工作。在此表示感谢！书中不足之处，敬请批评指正。



www.insight-ap.com

香港科汇（亚太）有限公司盈丰分部

香港：852-2410-2780
北京：86-10-85191859
成都：86-28-6199198
重庆：86-23-68790845
南京：86-25-4204221
上海：86-21-62159935
深圳：86-755-3664389
武汉：86-27-87322660
厦门：86-592-5163621
西安：86-29-8252934



www.xilinx.com

赛灵思公司

香港：852-24245200
上海：86-21-68862323