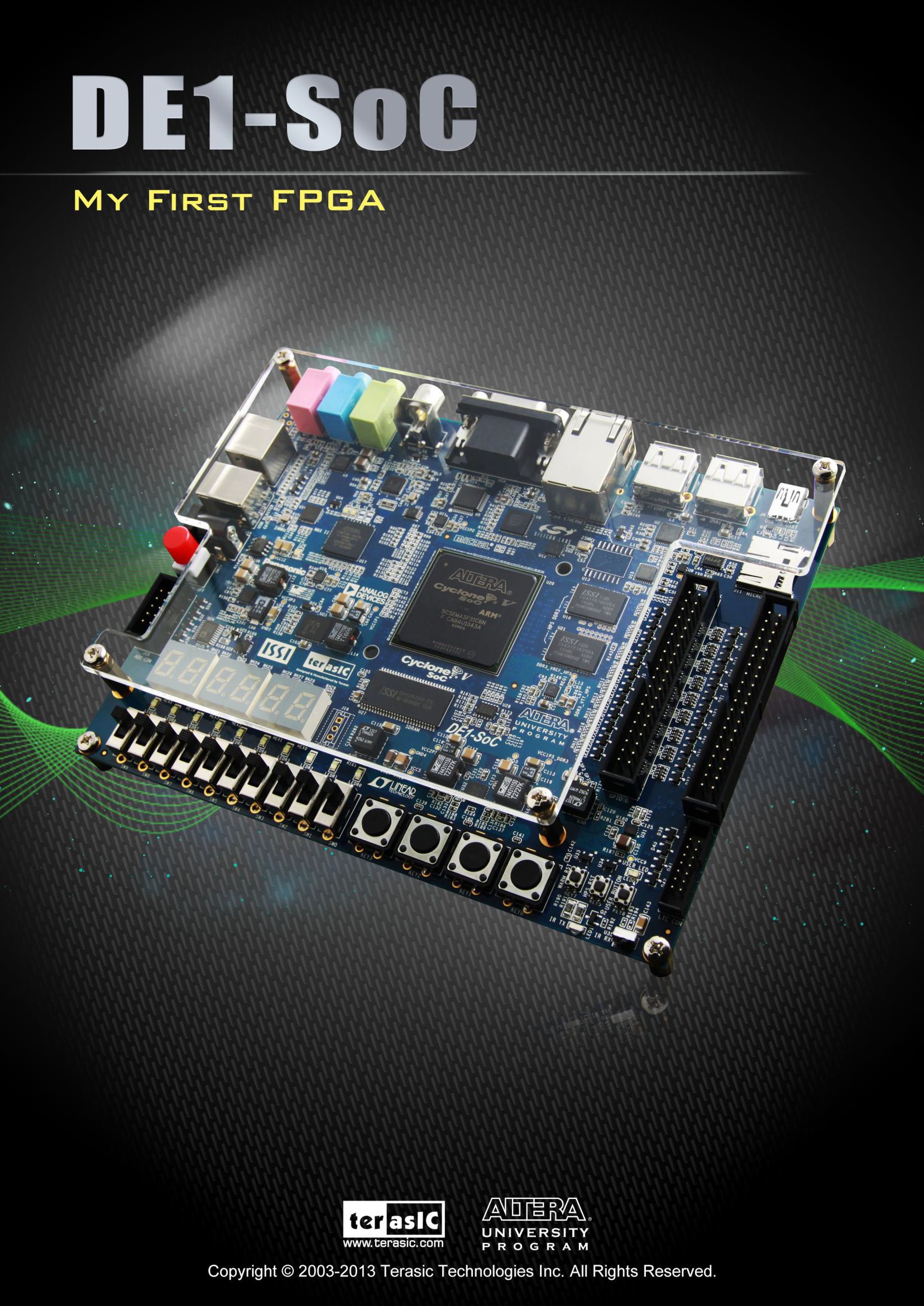


订阅DeepL Pro以编辑此文档。  
访问www.DeepL.com/Pro，了解更多信息。



***目 录***

[第1章 INTRODUCTION 1](#_bookmark0)

* 1. [设计流程1](#_bookmark1)
  2. [在你开始之前2](#_bookmark3)
  3. [你将学到什么6](#_bookmark8)

[第2章 指定的DEVICE 7](#_bookmark9)

[2.1 指派设备7](#_bookmark10)

[第三章 设计 ENTRY 11](#_bookmark14)

* 1. [添加一个Pll的Megafunction11](#_bookmark15)
  2. [添加一个多路复用器26](#_bookmark35)
  3. [指定引脚34](#_bookmark43)
  4. [创建一个默认的时间任务SDC文件36](#_bookmark45)

[第四章 编译和验证你的DESIGN 38](#_bookmark46)

* 1. [汇编你的设计38](#_bookmark47)
  2. [对FPGA设备进行编程40](#_bookmark50)
  3. [验证硬件44](#_bookmark54)

[第5章 APPENDIX 47](#_bookmark57)

[5.1 总部和分支机构47](#_bookmark58)

**第一章**

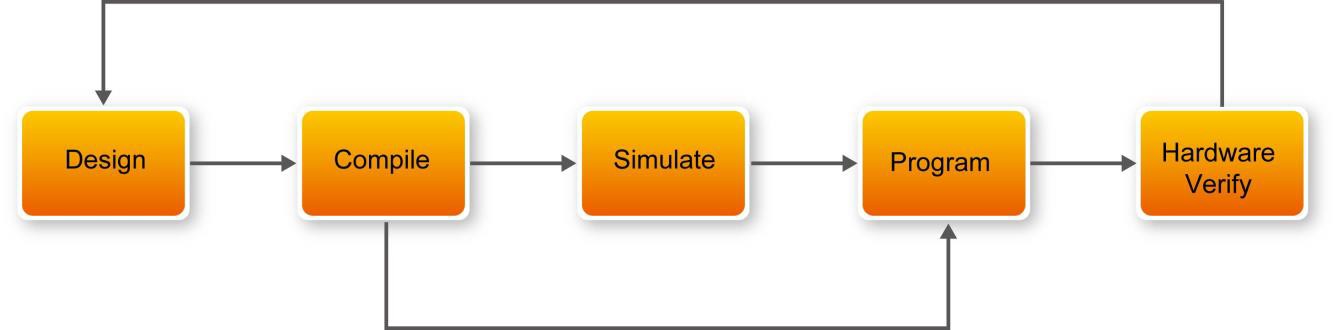
***简介***

本教程提供了全面的信息，将帮助你了解如何创建一个FPGA设计并在DE1-SoC开发板上运行它。下面几节提供了设计流程的快速概述，解释了你需要开始的内容，并描述了你将学习的内容。

## **设计流程**

[**图1-1**](#_bookmark2)显示了FPGA设计流程框图。

标准的FPGA设计流程始于使用原理图或硬件描述语言（HDL）的设计输入，如Verilog HDL或VHDL。在这个步骤中，你可以创建一个在FPGA内实现的数字电路。然后，该流程在FPGA硬件中进行编译、仿真、编程和验证。



### 图1-1 设计流程

本教程指导你完成除仿真以外的所有步骤。虽然本文件没有涉及，但模拟是非常重要的学习内容，而且有整个应用程序专门用于模拟硬件设计。有两种类型的仿真，即功能仿真和时序仿真，功能仿真允许你验证你的代码是否适当地操作了输入和输出。时序（或后置布局）仿真可以验证设计是否符合时序，以及在设备中是否有适当的功能。

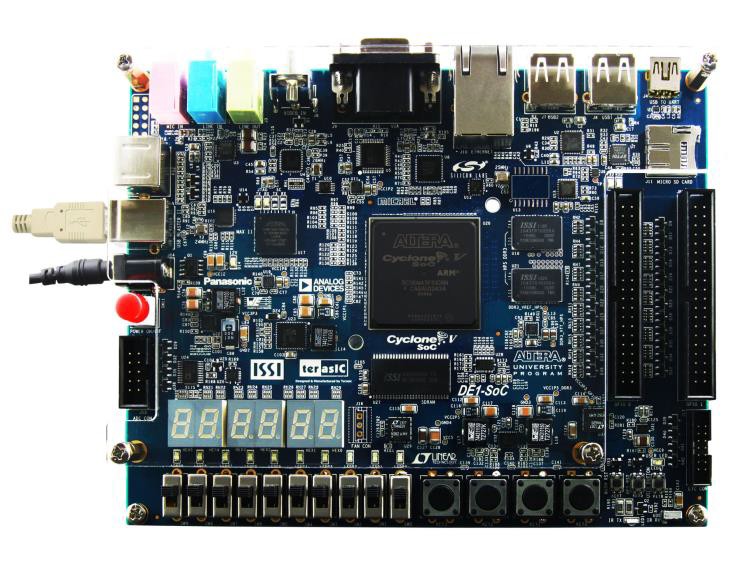
## **在你开始之前**

本教程假定有以下先决条件

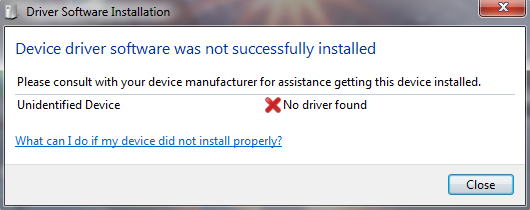
* 你一般知道什么是FPGA。本教程不解释可编程逻辑的基本概念。
* 你对数字电路设计和电子设计自动化（EDA）工具有一定的了解。
* 你已经在你的计算机上安装了Altera Quartus II 13.0软件。如果你没有Quartus II软件，你可以从Altera网站上下载，网址是[www.altera.com/download。](http://www.altera.com/download)
* 你有一块DE1-SoC开发板，你将在上面测试你的项目。使用开发板可以帮助你验证你的设计是否真的可行。
* 您已经阅读了快速入门指南和/或您的开发工具包的入门用户指南。这些文件确保你有。
* 安装了所需的软件。
* 确定开发板功能正常并与你的计算机连接。

下一步你应该安装USB-BlasterII驱动程序，插入12伏的适配器为板子提供电源。使用USB电缆将DE1-SoC板上最左边的USB接口连接到运行Quartus II软件的计算机上的USB端口。打开DE1-SoC板上的电源开关。

[**如图1-2**](#_bookmark4)所示，计算机将识别连接到其USB端口的新硬件并接通电路板的电源[，](#_bookmark4)但如果它没有已经安装的所需驱动程序，则无法继续。DE1-SoC板是通过使用Altera USB-Blaster II机制进行编程的。如果尚未安装USB-BlasterII驱动，将出现[**图1-3**](#_bookmark5)中的驱动软件安装。点击关闭。

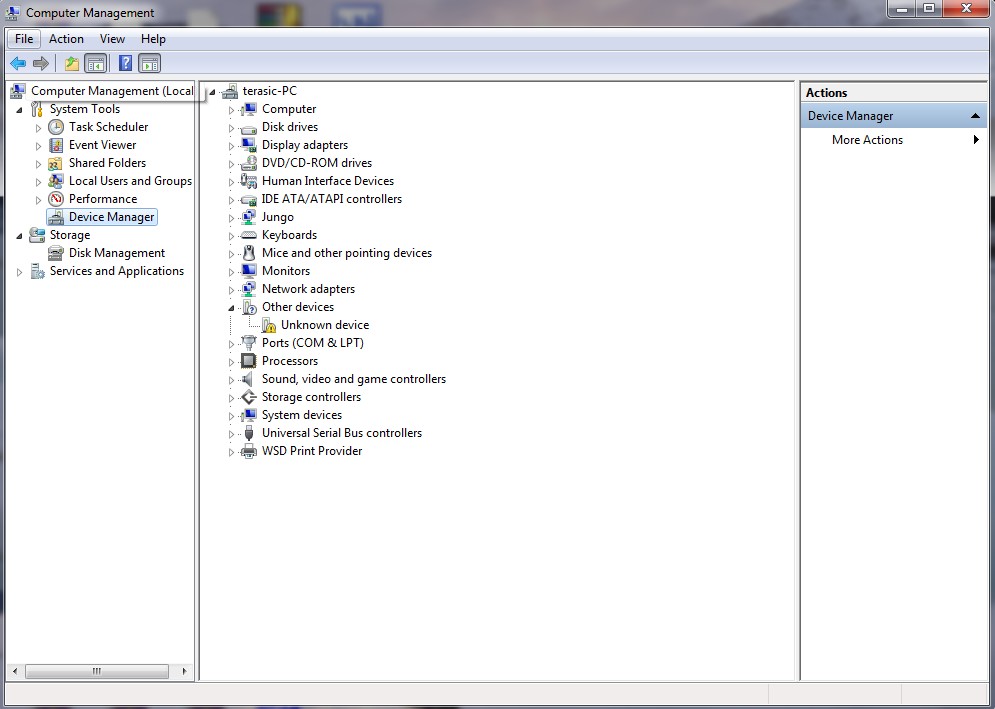


### 图1-2 连接设置



**图1-3 驱动程序软件安装**

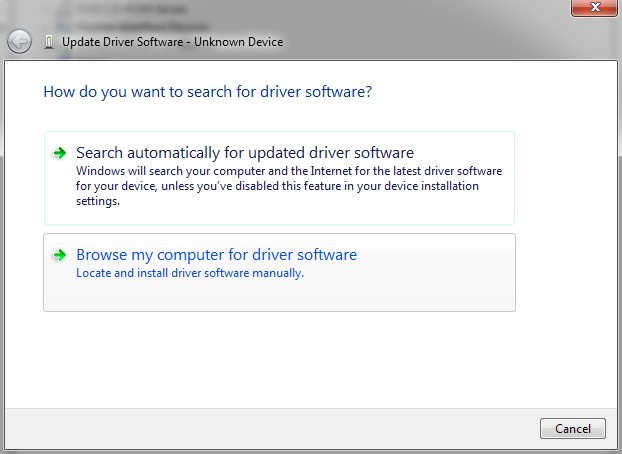
由于所需的驱动程序在Windows Update网站上不可用，所以打开计算机管理，选择设备管理器。这就导致了[**图1-4**](#_bookmark6)中的窗口[。](#_bookmark6)



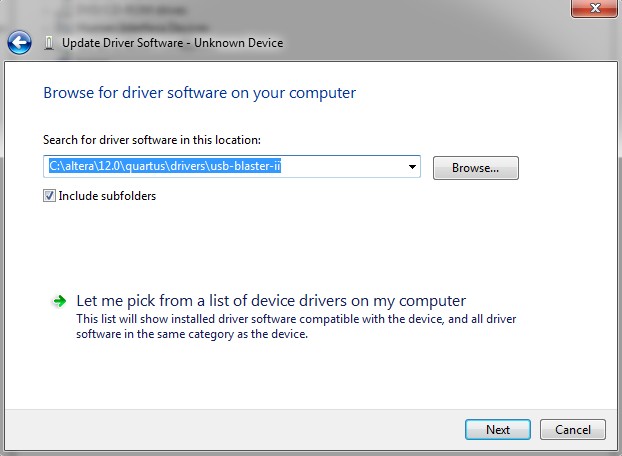
### 图1-4 设备管理器

右键单击 "其他设备">"未知设备"，并选择 "更新驱动软件"......这将导致出现以下窗口

### 图1-5

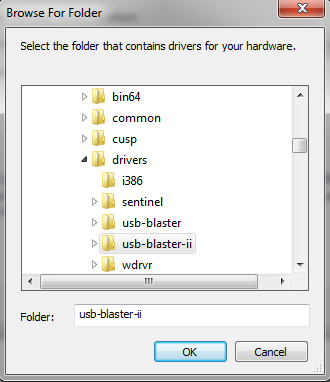


**图1-5 更新驱动软件**

驱动程序可以在Quartus II软件中找到。因此，点击浏览我的电脑，寻找设备软件，进入**图1-6。**

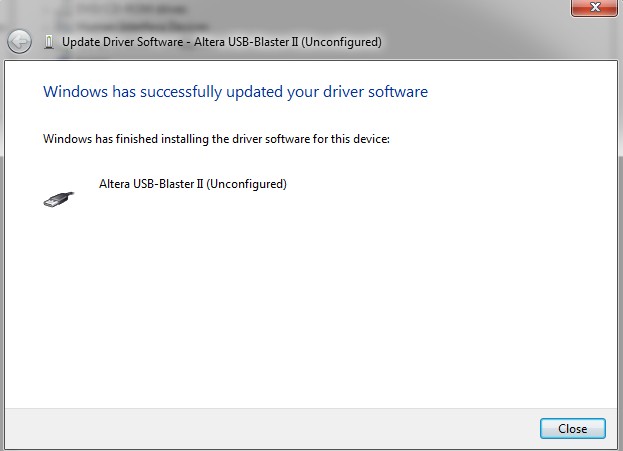
**图1-6 指定驱动程序的位置**

现在，点击浏览，在**图1-7**的弹出框中找到所需的驱动程序，它的位置是C:\altera\13.0\quartus\drivers\usb-blaster-ii。点击确定，然后在回到图1-6时，点击下一步。



### 图1-7 浏览查找位置

现在驱动程序将被安装，[**如图1-6所示。**](#_bookmark7)点击关闭，你就可以开始使用DE1-SoC板了。



### 图1-6 驱动程序已安装

## **你将学到什么**

在本教程中，你将执行以下任务。

创建一个设计，使开发板上的LED以输入键控制的速度闪烁--这个设计很容易创建，并给你视觉反馈，说明设计是有效的。当然，你也可以使用你的DE1-SoC板来运行其他设计。对于LED设计，你将为一个简单的32位计数器编写Verilog HDL代码，添加一个锁相环（PLL）大函数作为时钟源，并添加一个2输入复用器大函数。当设计在电路板上运行时，你可以按下一个输入开关来复用驱动输出LED的计数器位。

熟悉Quartus II设计工具-本教程不会使你成为专家（请参考DE1-SoC tut\_quartus\_intro\_verilog文件），但在最后，你将了解Quartus II项目的基本概念，如使用原理图编辑器和HDL输入设计，编译你的设计，并将其下载到DE1-SoC开发板的FPGA中。

为学习更多关于FPGA的知识打下基础-例如，您可以在单个芯片上创建并下载数字信号处理（DSP）功能，或者建立一个多处理器系统，或者在同一芯片上创建任何您可以想象的其他功能。你不需要翻阅资料书来寻找完美的逻辑器件或创造你自己的ASIC。你所需要的只是你的计算机、你的想象力和一块Altera DE1-SoC FPGA开发板。

**第二章**

***指派设备***

在本教程中，您首先创建一个新的Quartus II项目。项目是一组维护FPGA设计信息的文件。Quartus II设置文件（.qsf）和Quartus II项目文件（.qpf）文件是Quartus II项目中的主要文件。要编译设计或进行引脚分配，你必须首先创建一个项目。

## **2.1 指派设备**

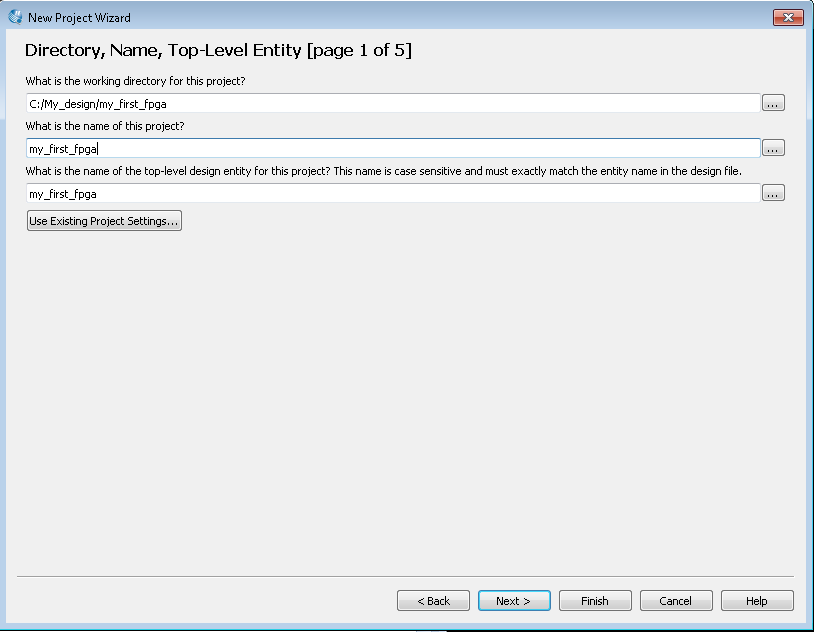
1. 在Quartus II软件中，选择File > New Project Wizard。打开 "介绍 "页面。参见

### 图2-1

**图2-1 新项目向导介绍**

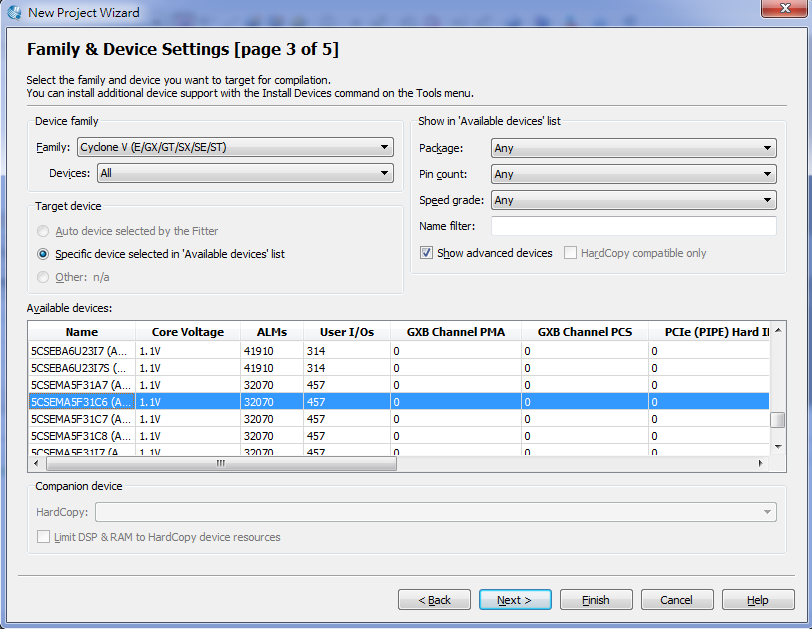
1. 点击下一步。
2. 输入关于你的项目的以下信息。
   1. 这个项目的工作目录是什么？请输入一个目录，您将在该目录中存储本设计的Quartus II项目文件。
   2. 例如，C:\My\_design\my\_first\_fpga。
   3. Quartus II软件中的文件名、项目名和目录不能包含空格。
   4. 这个项目的名称是什么？输入my\_first\_fpga。
   5. 这个项目的顶层设计实体的名称是什么？输入my\_first\_fpga。请看

### [图2-2.](#_bookmark11)



**图2-2 项目信息**

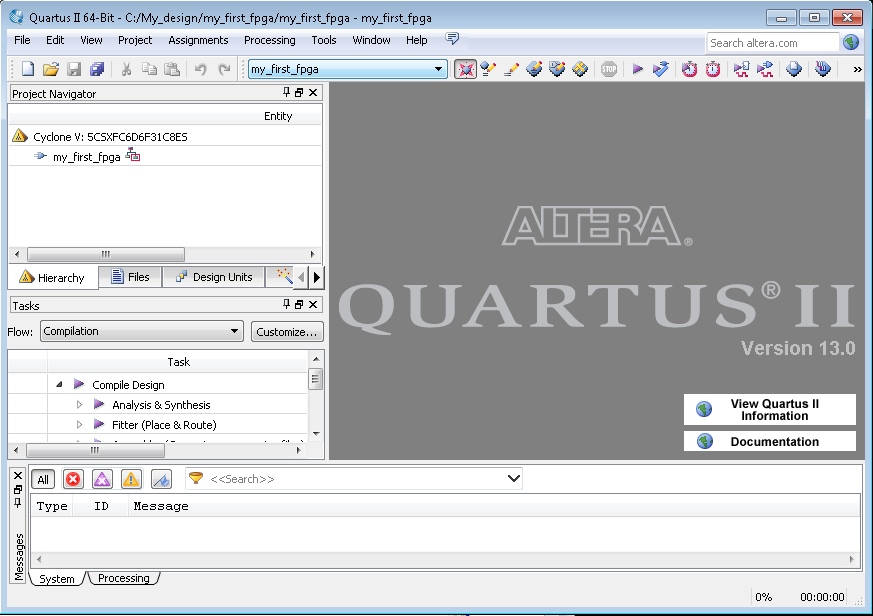
* 1. 点击下一步。
  2. 你将为设计分配一个特定的FPGA器件并进行引脚分配。[**见图2-3**。](#_bookmark12)



### 图2-3 指定设备示例

* 1. 单击 "完成"。

1. 当提示时，选择Yes来创建my\_first\_fpga项目目录。你刚刚创建了你的第一个Quartus II FPGA项目。[**见图2-4**](#_bookmark13)。



### 图2-4 my\_first\_fpga项目

**第三章**

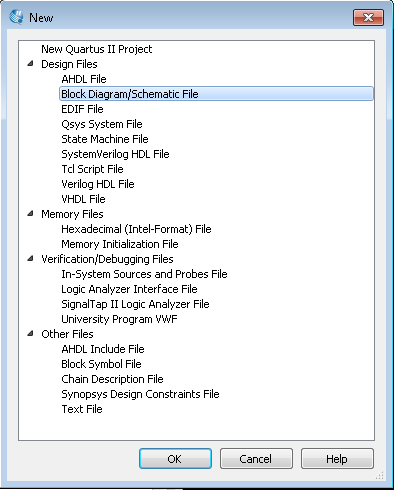
***设计条目***

## **添加一个PLL大函数**

本节介绍了如何添加一个PLL大函数

在设计输入步骤中，你创建一个原理图或块设计文件（.bdf），这是顶层设计。你将添加参数化模块库（LPM）功能，并使用Verilog HDL代码来添加逻辑块。当创建你自己的设计时，你可以选择这些方法中的任何一种或它们的组合。

1. 选择 "文件">"新建">"框图/化学文件"（见图[**3-1），**](#_bookmark16)创建一个新文件，即Block1.bdf，你将把它保存为顶层设计。

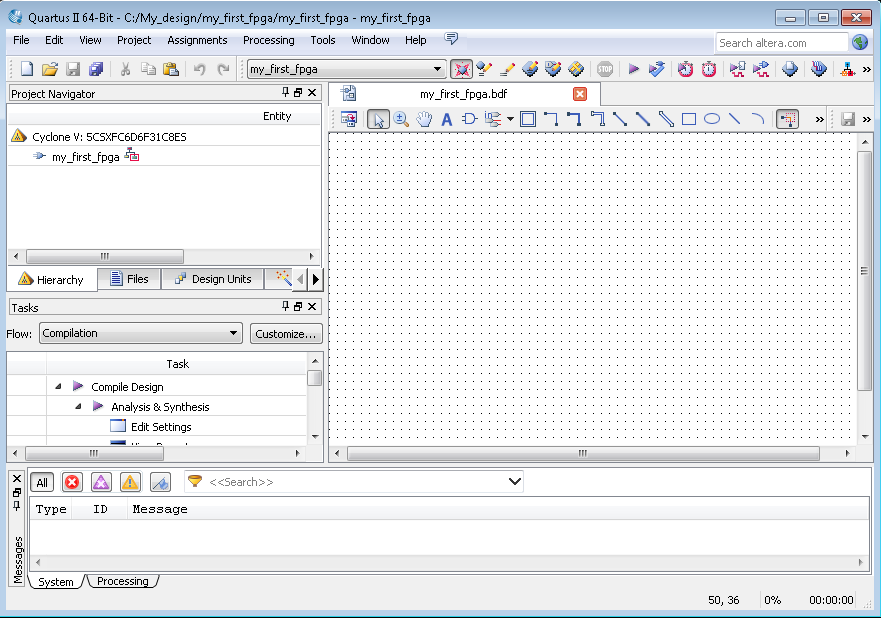


### 图3-1 新的BDF

1. 单击 "确定"。
2. 选择文件>另存为，并输入以下信息。

* 文件名：my\_first\_fpga
* 保存为类型。框图/示意图文件（\*.bdf）。

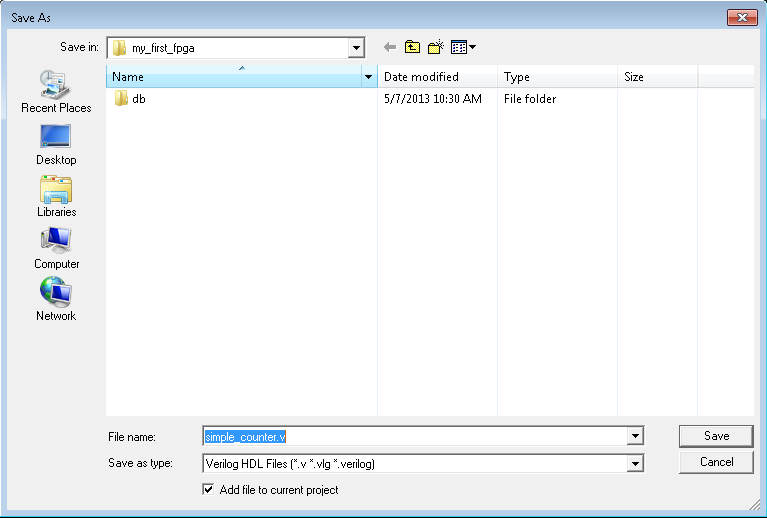
1. 点击保存。新的设计文件出现在块编辑器中（[**见图3-2**](#_bookmark17)）。



### 图3-2 银行BDF

1. 通过选择文件>新建>Verilog HDL文件将HDL代码添加到空白框图中。
2. 点击 "确定 "创建一个新文件Verilog1.v，你将把它保存为simple\_counter.v。
3. 选择 "文件">"另存为 "并输入以下信息（[**见图3-3）。**](#_bookmark18)

* 文件名： simple\_counter.v
* 保存为类型。Verilog HDL文件（\*.v, \*.vlg, \*.verilog）。



### 图3-3 保存Verilog HDL文件

产生的空文件已经准备好让你输入Verilog HDL代码。

1. 在空白的simple\_counter.v文件中键入以下Verilog HDL代码（[**见图3-4**](#_bookmark19) simple\_counter.v的Verilog文件）。

//它有一个单一的时钟输入和一个32位的输出端口模块 simple\_counter (

CLOCK\_50。

counter\_out

);

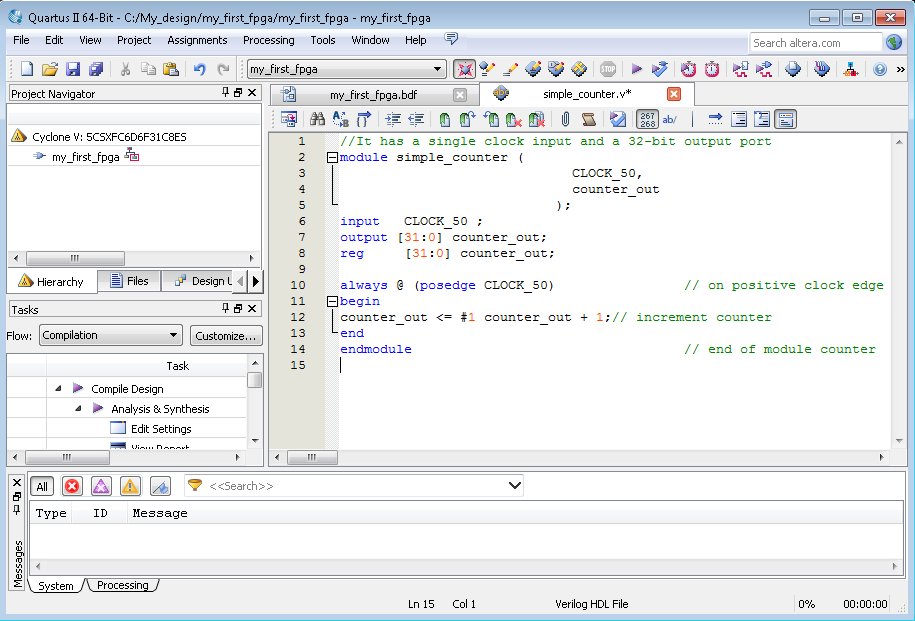
input CLOCK\_50 ; 输出 [31:0] counter\_out。

reg [31:0] counter\_out;

始终 @ (posedge CLOCK\_50) // 在正的时钟边沿开始

counter\_out <= #1 counter\_out + 1;//增加计数器 end

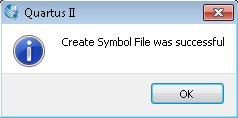
endmodule // 模块的计数器endmodule 结束



### 图3-4 simple\_counter.v的Verilog文件

1. 选择 "文件">"保存"，按Ctrl + s，或点击软盘图标来保存文件。
2. 选择文件>创建/更新>为当前文件创建符号文件，将simple\_counter.v文件转换为符号文件（.sym）。你使用这个符号文件将HDL代码添加到你的BDF原理图。

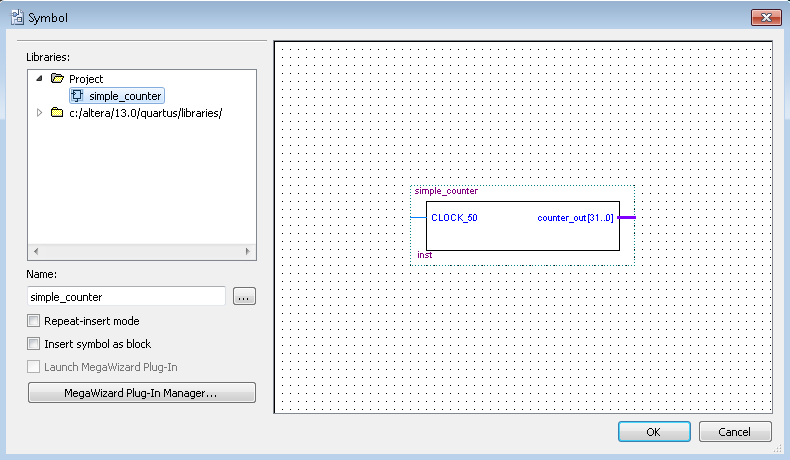
Quartus II软件创建一个符号文件并显示一条信息（[**见图3-5）**](#_bookmark20)。



### 图3-5 创建符号文件已成功

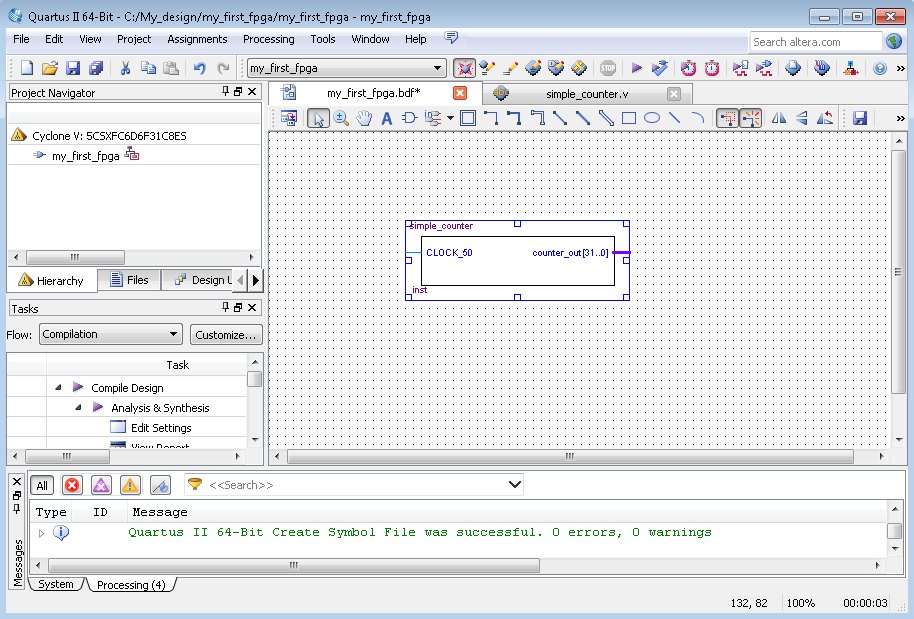
1. 单击 "确定"。
2. 要将simple\_counter.v符号添加到顶层设计中，点击my\_first\_fpga.bdf标签。
3. 选择编辑>插入符号。
4. 双击项目目录以展开它。
5. 点击新创建的simple\_counter符号，选择它的图标。

你也可以在BDF的空白区域双击来打开符号对话框



### 图3-6 将符号添加到BDF中

1. 单击 "确定"。
2. 将光标移到BDF网格上；符号图像随光标移动。点击将simple\_counter符号放到BDF上。你可以在放置后移动该块，只需点击并拖动它到你想要的位置，然后释放鼠标按钮来放置它。[**见图3-7**。](#_bookmark21)



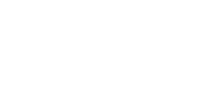
### 图3-7 放置 simple\_counter 符号

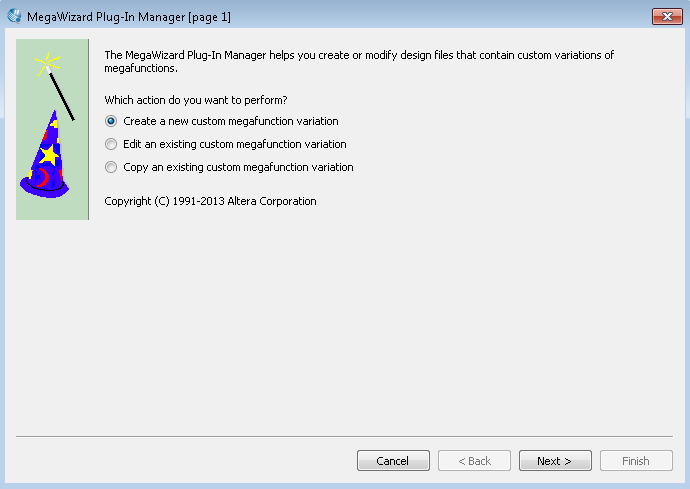
1. 按Esc键或点击原理图网格上的一个空位，可以取消进一步放置这个符号的实例。
2. 定期保存你的项目。

使用Quartus添加PLL大型函数

巨型函数，如LPM中提供的那些，是预先设计的模块，你可以在FPGA设计中使用。这些Altera提供的Megafunctions针对速度、面积和器件系列进行了优化。您可以增加

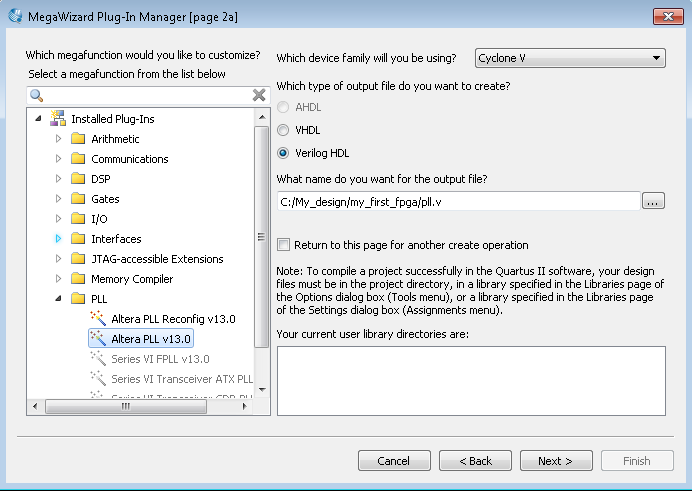
通过使用巨型函数而不是自己编写函数来提高效率。Altera还提供更复杂的函数，称为MegaCore函数，你可以免费评估这些函数，但在生产设计中使用时需要许可证文件。本教程设计使用一个PLL时钟源来驱动一个简单的计数器。PLL使用板载振荡器（DE1-SoC板为50MHz）来创建一个恒定的时钟频率作为计数器的输入。为了创建时钟源，你将添加一个预先建立的名为Altera PLL的LPM大功能。

1. 选择 "编辑">"插入符号 "或点击工具栏上的 "添加符号"------。
2. 点击Megawizard Plug-in Manager。MegaWizard® 插件管理器出现（[**见图3-8**](#_bookmark22)）。



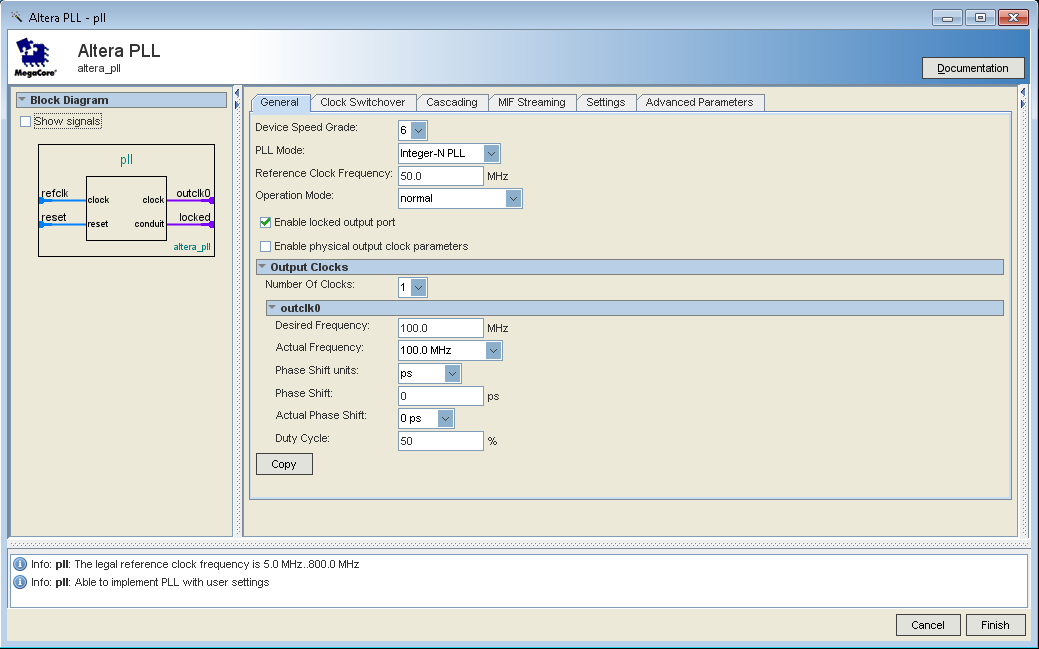
### 图3-8 Mega Wizard插件管理器

1. 点击下一步。
2. 在MegaWizard插件管理器[第2a页]，指定以下选择（[**见图3-9**](#_bookmark23)）。
   1. 选择PLL>Altera PLL v13.0。
   2. 在你将使用哪个器件系列？选择Cyclone V for DE1-SoC开发板。
   3. 在你想创建哪种类型的输出文件？选择Verilog HDL。
   4. 在你想给输出文件取什么名字？在已经创建的目录名后面输入pll。
   5. 点击下一步。



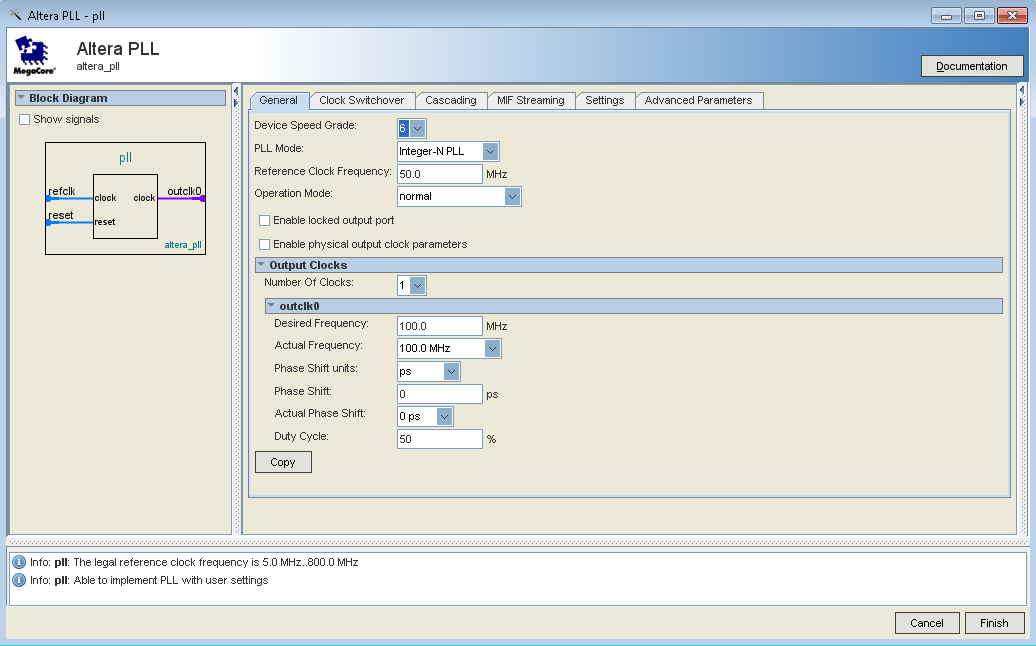
### 图3-9 MegaWizard 插件管理器 [第2a页] 选择

1. 在MegaWizard插件管理器窗口中，进行以下选择（[**见图3-10**](#_bookmark24)）。
   1. DE1-SoC的设备速度等级选择6。
   2. 设置Refclk输入的频率为50MHz。



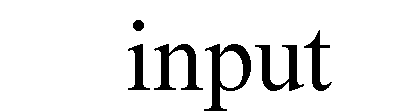
### 图3-10 MegaWizard插件管理器的选择

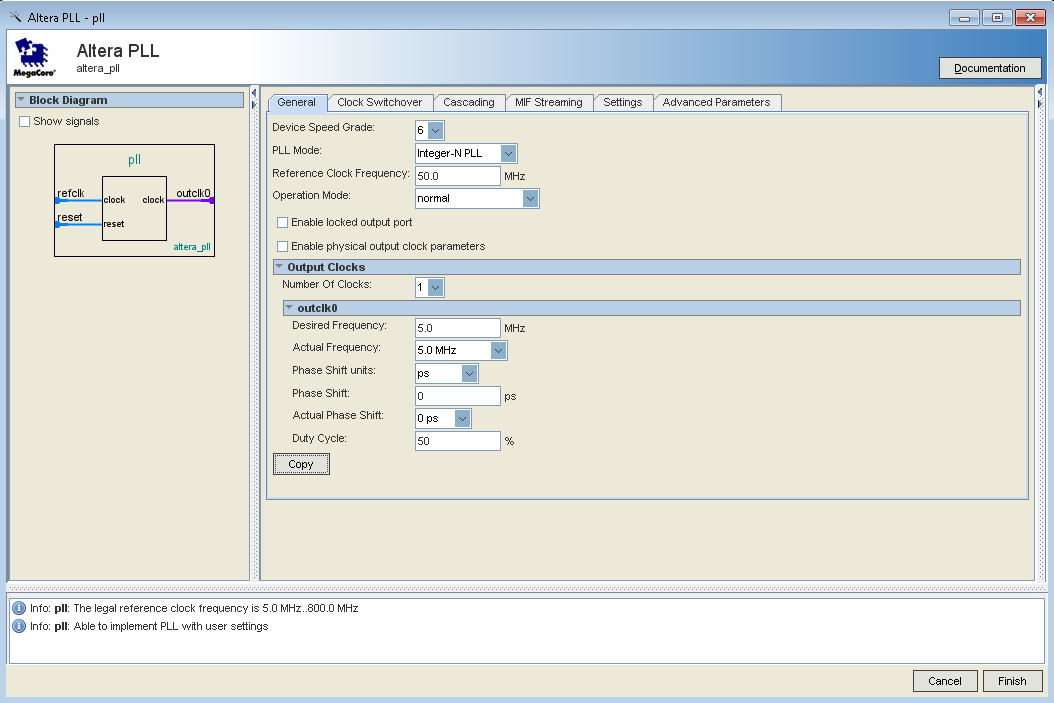
1. 关掉MegaWizard上的启用锁定输出端口选项。当你关闭它时，引脚会从PLL块的图形预览中消失。[**见图3-11**](#_bookmark25)的例子。



### 图3-11 MegaWizard插件管理器的选择

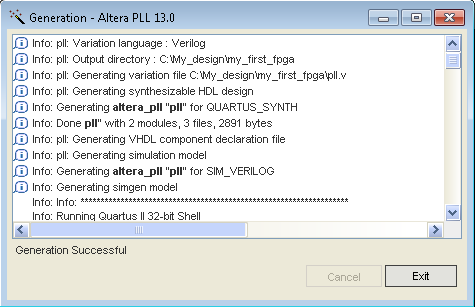
1. 然后，进行以下选择（[**见图3-12**](#_bookmark26)）。

希望的Frequency 5.0



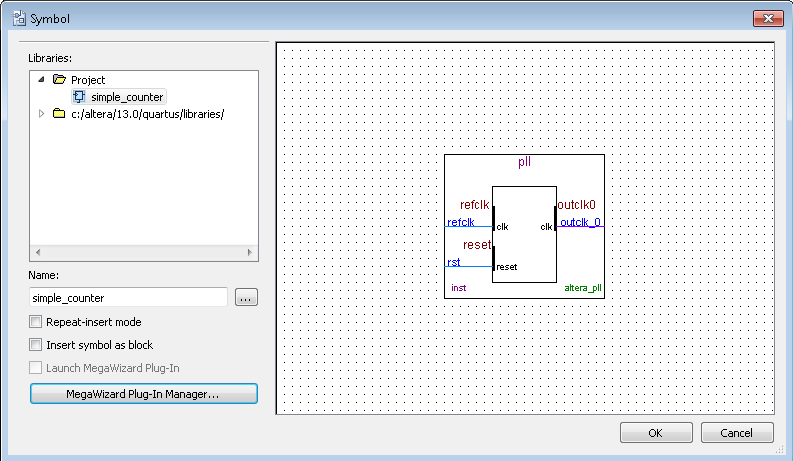
### 图3-12 MegaWizard插件管理器的选择

1. 点击完成。这将导致**图3-13.**点击退出。



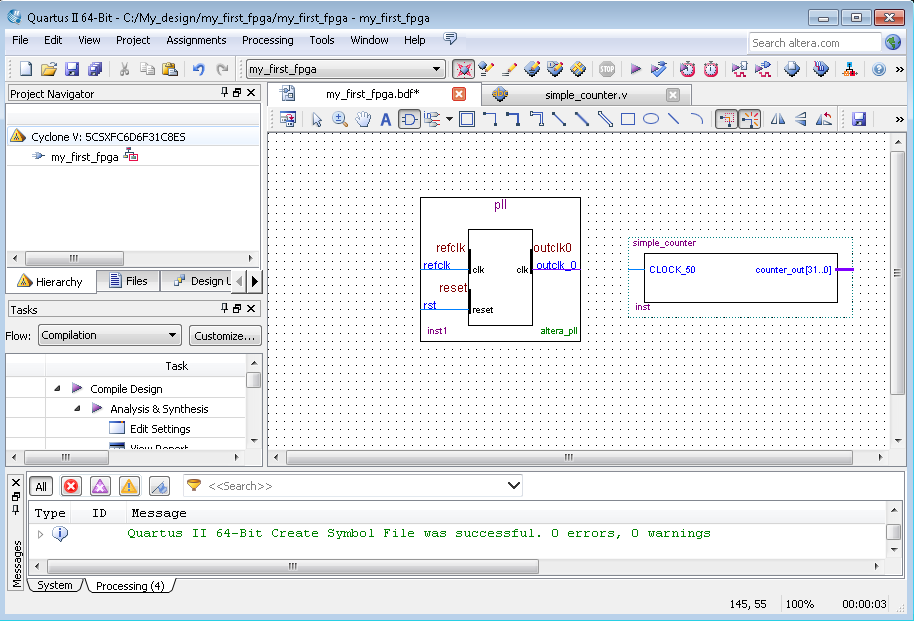
### 图3-13 PLL生成

符号窗口打开，显示新创建的PLL大函数。[**见图3-14**。](#_bookmark27)



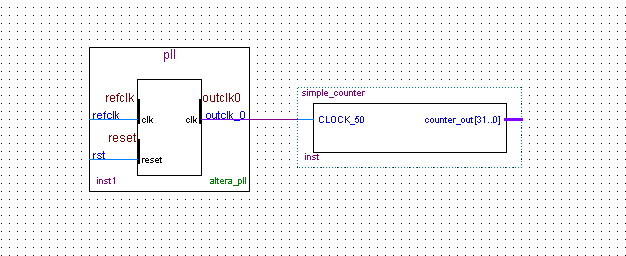
### 图3-14 PLL符号

1. 点击 "确定"，将pll符号放到BDF上，放在simple\_counter符号的左边。你可以通过按住鼠标左键来移动符号，帮助你确保它们正确排队。[**见图3-15**。](#_bookmark28)



### 图3-15 放置PLL符号

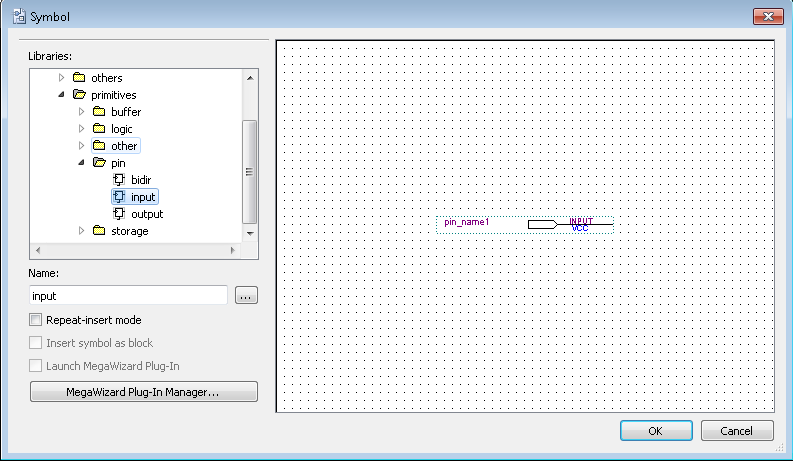
1. 移动鼠标，使光标（也称为选择工具）位于pll符号的outclk\_0输出引脚上。正交节点工具（十字准星）图标出现。
2. 点击并拖动一条从outclk\_0输出到simple\_counter时钟输入的总线线。这个动作将pll输出与simple\_counter输入绑在一起（[**见图3-16**](#_bookmark29)）。



### 图3-16 画一条总线连接pll outclk\_0端口到simple\_counter CLOCK\_50端口

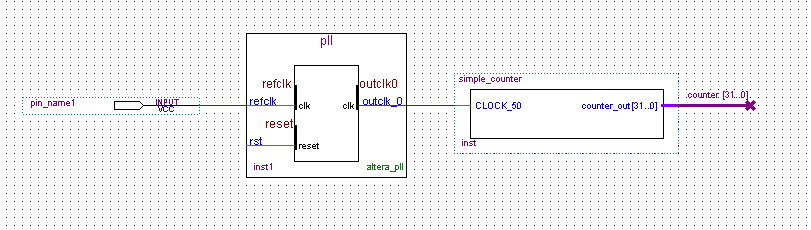
1. 通过以下步骤添加一个输入引脚和一个输出总线。
2. 选择编辑>插入符号。
3. 在库下，选择quartus/libraries > primitives > pin >input。[**见图3-17**](#_bookmark30)
4. 单击 "确定

如果你需要更多的空间来放置符号，你可以使用BDF窗口边缘的垂直和水平滚动条来查看更多的绘图空间。



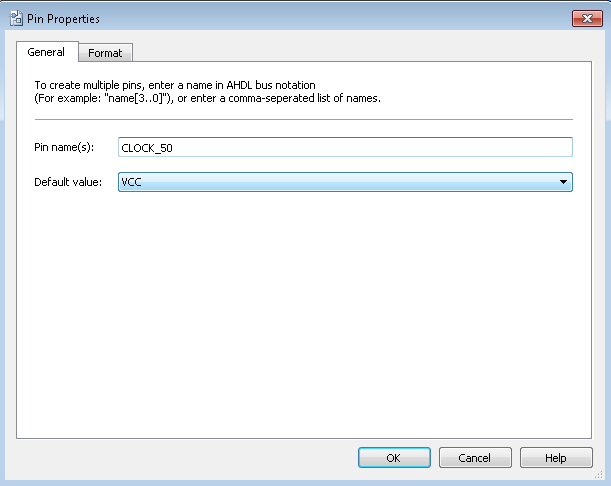
### 图3-17 输入引脚符号

1. 将新的针脚放在BDF上，使其接触到pll符号的输入。
2. 用鼠标点击并将新的输入引脚拖到左边；注意，端口仍然是连接的，[**如图3-18**](#_bookmark31)所示[。](#_bookmark31)



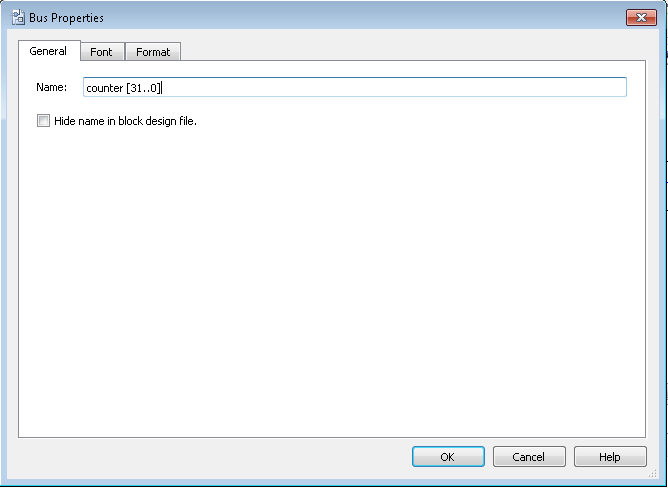
### 图3-18 连接PLL符号和输入端口

1. 通过双击pin\_name并输入CLOCK\_50来改变引脚名称（[**见图3-19**](#_bookmark32)）。这个名称与连接到FPGA的振荡器时钟相关。
2. 使用正交总线工具，画一条总线，一边连接到simple\_counter的输出端口，另一端在simple\_counter右边大约4到8个网格空间处不连接。

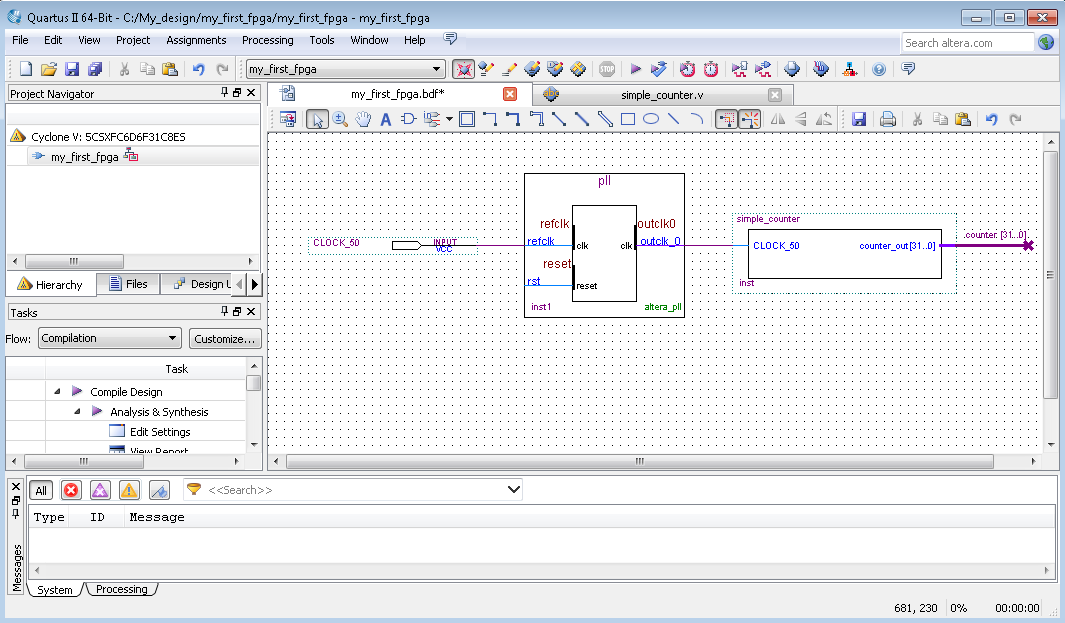


### 图3-19 更改输入端口名称

1. 右键单击新的输出总线线，选择属性。
2. 使用正交总线工具，画一条总线，一边连接到simple\_counter的输出端口，另一端在simple\_counter右边大约6到8个网格空间处不连接。
3. 输入计数器[31...0]作为总线名称（[**见图3-20**](#_bookmark33)）。符号[X ...Y]是Quartus II在BDF原理图中指定总线宽度的方法，其中X是最有效位（MSB），Y是最小有效位（LSB）。
4. 单击 "确定"。[**图3-21**](#_bookmark34)显示了BDF的情况。



### 图3-20 更改输出BUS名称

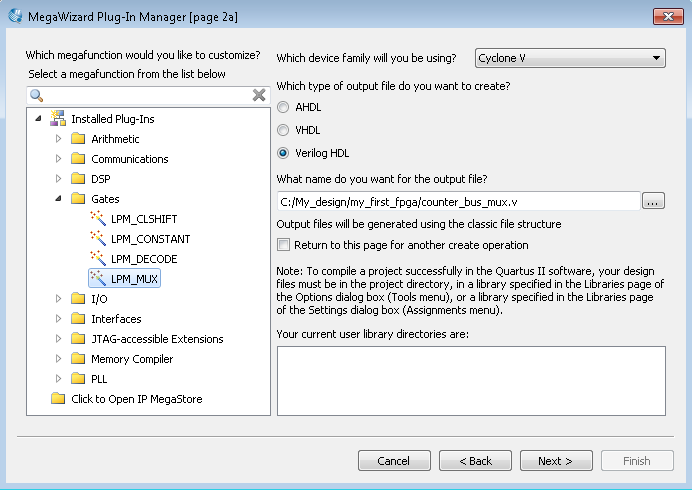


**图3-21 BDF**

## **添加一个多路复用器**

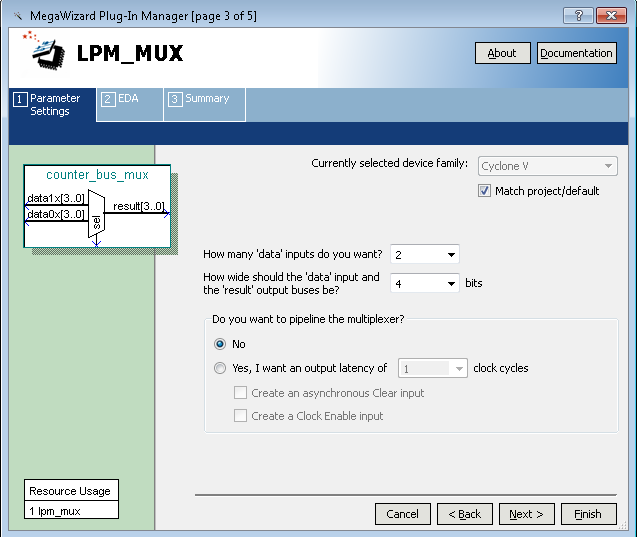
本设计使用一个多路复用器将simple\_counter的输出路由到DE1-SoC开发板上的LED引脚。你将使用MegaWizard插件管理器来添加多路复用器lpm\_mux。该设计将计数器总线的两种变化复用到DE1-SoC开发板的四个LED上。

1. 选择编辑>插入符号。
2. 点击Megawizard Plug-in Manager。
3. 点击下一步。
4. 选择已安装的插件 > 盖茨 > LPM\_MUX。
5. 选择与你正在使用的开发板上的器件对应的器件系列，选择Verilog HDL作为输出文件类型，并将输出文件命名为counter\_bus\_mux.v（[**见图3-22**](#_bookmark36)）。
6. 点击下一步。



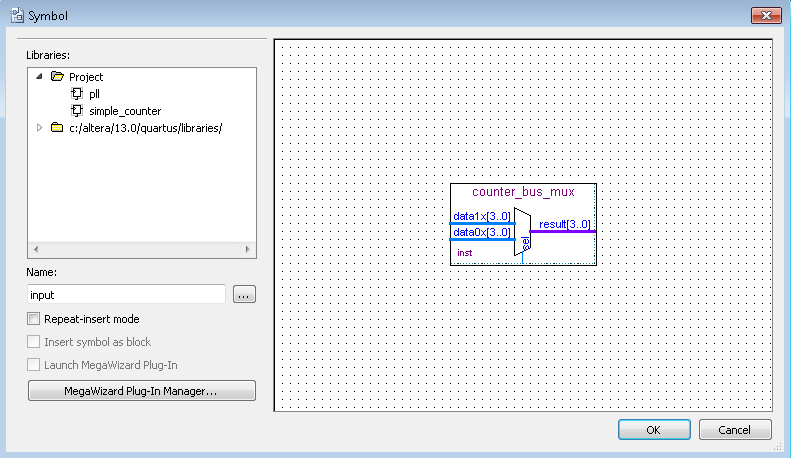
### 图3-22 选择lpm\_mux

1. 在你想要多少个 "数据 "输入？选择2个输入（默认）。
2. 在 "数据输入和结果输出应该有多'宽'？选择4（[**见图3-23**](#_bookmark37)）。



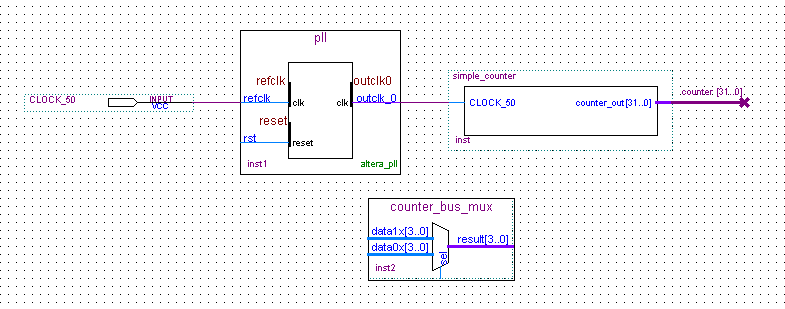
### 图3-23 lpm\_mux设置

1. 点击下一步。
2. 点击完成两次。出现符号窗口（[**见图3-24**](#_bookmark38)的例子）。



### 图3-24 lpm\_mux 符号

1. 单击 "确定
2. 将counter\_bus\_mux符号放在BDF上现有符号的下面。[**见图3-25**](#_bookmark39)

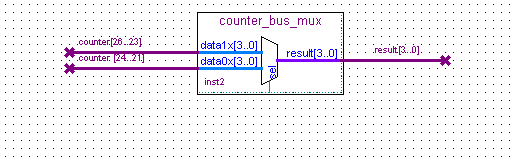


### 图3-25 放置lpm\_mux符号

1. 在counter\_bus\_mux符号中添加输入总线和输出引脚，如下所示。
   1. 使用正交总线工具，从data1x[3..0]和data0x[3..0]输入端口到counter\_bus\_mux左边大约8到12个网格空间绘制总线线。
   2. 从结果[3...0]输出端口到counter\_bus\_mux右边的大约4到8个网格空间画一条总线线。
   3. 右键单击连接到data1x[3..0]的总线线路，选择属性。
   4. 命名总线计数器[26...23]，它只选择那些计数器的输出位来连接到data1x输入的四个位。

因为counter\_bus\_mux的输入总线与simple\_counter的输出总线名称相同（counter[x ... y]），所以Quartus II软件知道要连接这些总线。

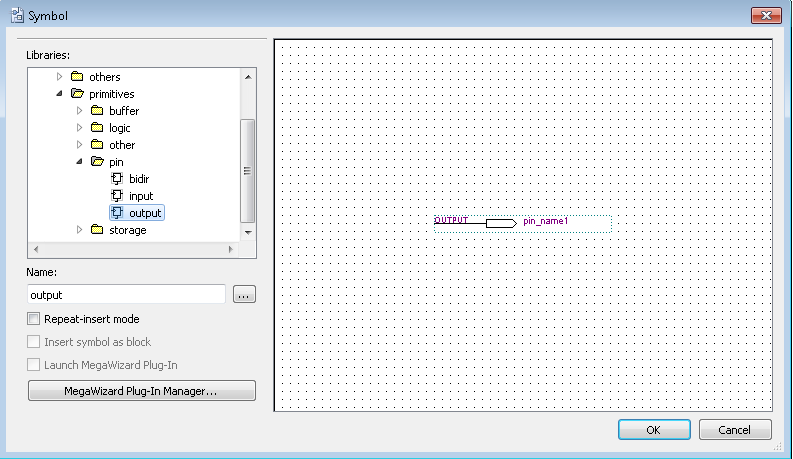
* 1. 单击 "确定"。
  2. 右键单击连接到data0x[3..0]的总线线路，选择属性。
  3. 命名总线计数器[24...21]，它只选择那些计数器的输出位来连接到data1x输入的四个位。
  4. 单击 "确定"。[**图3-26**](#_bookmark40)显示了重新命名的总线。



### 图3-26 重新命名的 counter\_bus\_mux 总线

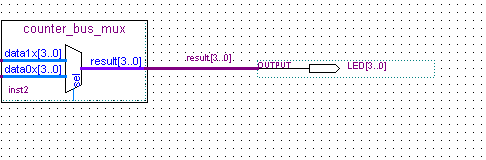
如果你还没有这样做，在继续之前保存你的项目文件。

1. 选择编辑>插入符号。
2. 在库下，双击quartus/libraries/ > primitives > pin > output（见图3-[**27）。**](#_bookmark41)



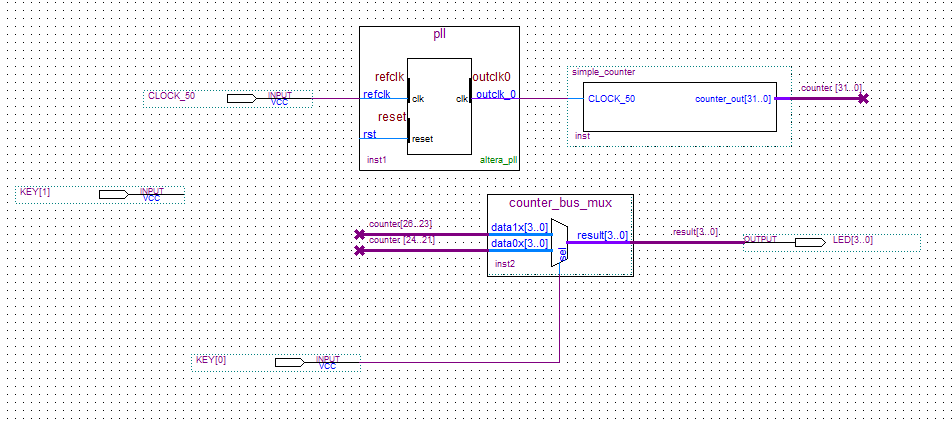
### 图3-27 选择一个输出引脚

1. 单击 "确定"。
2. 放置这个输出引脚，使其连接到 counter\_bus\_mux 结果 [3..0] 总线输出线。
3. 如步骤13 c和d所述，将输出引脚重命名为LED [3..0]（[**见图3-28**](#_bookmark42)）。

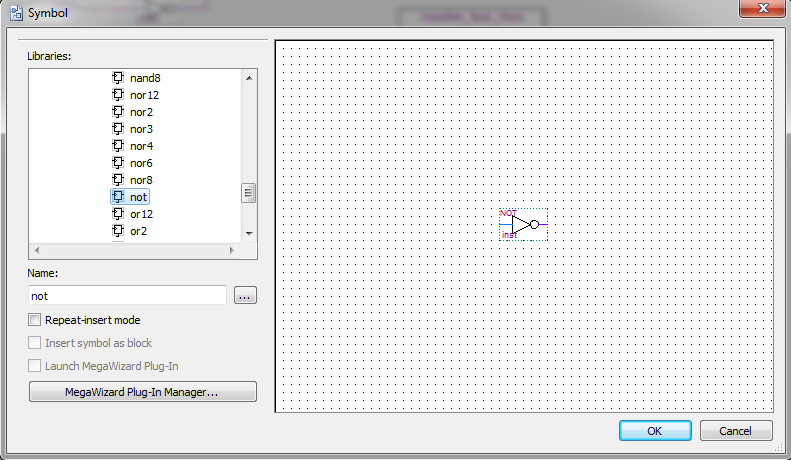


### 图3-28 重命名输出引脚

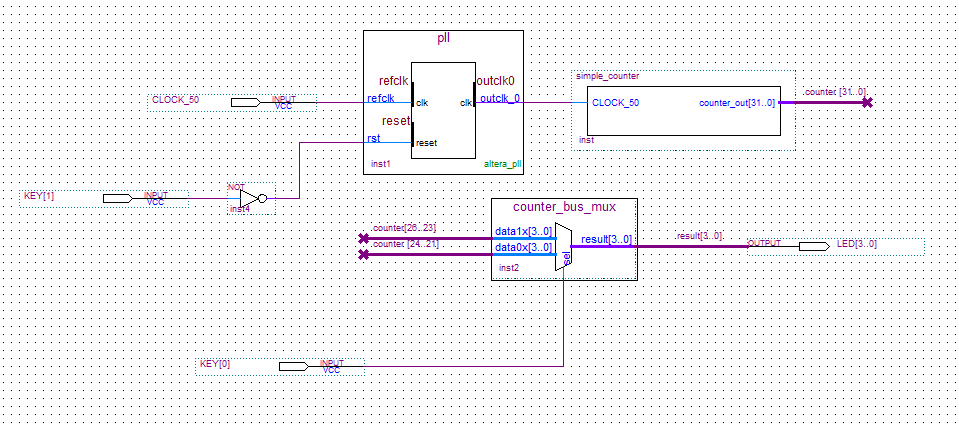
1. 使用输入引脚将一个输入引脚连接到多路复用器选择线。
   1. 选择编辑>插入符号。
   2. 在库下，双击quartus/libraries/ > primitives > pin > input。
   3. 单击 "确定"。
2. 把这个输入引脚放在 counter\_bus\_mux 下面。
3. 将输入引脚连接到 counter\_bus\_mux sel 引脚。
4. 重命名输入引脚为KEY [0]。
5. 在pll的左边放置另一个输入引脚，并将其重新命名为KEY[1]（**见图3-29**）。
6. 选择编辑>插入符号。
7. 在库下，双击quartus/libraries/ > primitives >logic > not（见图3-**30）。**
8. 单击 "确定"。
9. 放置这个不是引脚，使其与KEY[1]和pll rst相连接（**见图3-31）**。



### 图3-29 添加KEY [1:0] 输入引脚



**图3-30 选择一个不是引脚**



**图3-31 放置不是针**

你已经完成了向你的设计添加符号的工作。您可以使用工具栏上的文本工具（用A符号表示）将注释或信息作为文本添加到项目中。例如，你可以在KEY[0]输入引脚上添加标签 "OFF = SLOW, ON = FAST"，并添加项目描述，如 "我的第一个FPGA项目"。

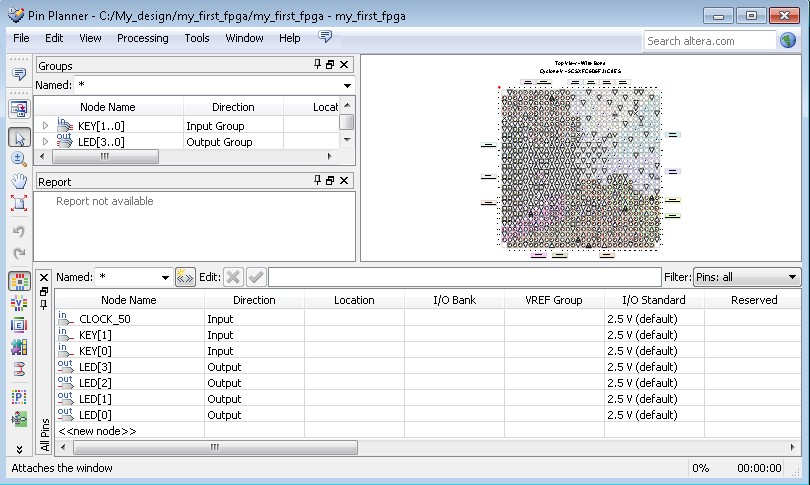
## **指派引脚**

在本节中，你将进行引脚分配。在进行引脚分配之前，请执行以下步骤。

1. 选择 "处理">"开始">"开始分析和阐述"，准备分配针脚位置。
2. 在分析和阐述完成后出现的信息窗口中点击确定。

要进行与KEY[1:0]和CLOCK\_50输入引脚以及LED[3.0]输出引脚相关的引脚分配，请执行以下步骤。

1. 选择 "分配">"引脚"，打开 "引脚规划器"，这是一个类似电子表格的具体引脚分配表。引脚计划表显示了设计的六个引脚。**见图3-32**



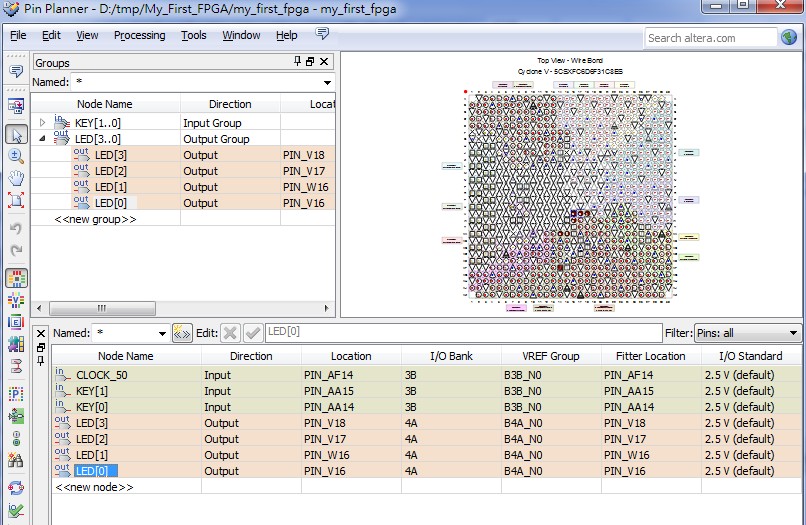
### 图3-32 引脚规划器示例

1. 在七个节点名称旁边的位置栏中，加入[**表3-1**](#_bookmark44)中所示的坐标（引脚编号），以获得与DE1-SoC板一起使用的实际值。

### 表3-1 引脚信息设置

|  |  |  |
| --- | --- | --- |
| ***引脚名称*** |  | ***FPGA引脚位置*** |
| **CLOCK\_50** | **AF14** | |
| **KEY[1]** | **AA15** | |
| **关键[0]** | **AA14** | |
| **LED[3]** | **V18** | |
| **LED[2]** | **V17** | |
| **LED[1]。** | **W16** | |
| **LED[0]。** | **V16** | |

双击七个引脚中的任何一个引脚的位置栏，打开一个下拉列表，输入表格中显示的位置，或者，你可以从下拉列表中选择引脚。例如，如果你输入F1并按下回车键，Quartus II软件会为你填入完整的PIN\_F1位置名称。该软件还跟踪相应的FPGA数据，如I/O库和VREF组。每个库都有一个明显的颜色，与右上角窗口中的顶视图线束图相对应。**见图3-33**。



### 图3-33 已完成的引脚规划实例

现在，你已经完成了你的Quartus II设计的创建。

## **创建一个默认的TimeQuest SDC文件**

时序设置对于一个成功的设计至关重要。在本教程中，你将创建一个基本的Synopsys设计约束文件（.SDC），Quartus II TimeQuest定时分析器在设计编译过程中会使用该文件。对于更复杂的设计，你将需要更仔细地考虑时序要求。

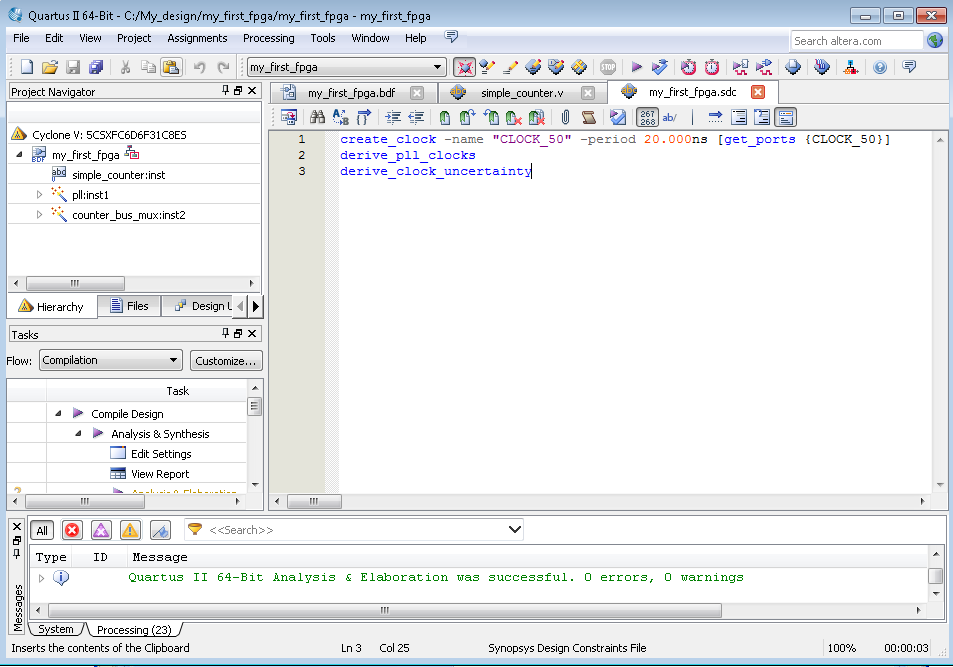
要创建一个SDC，请执行以下步骤。

1. 选择 "工具">"TimeQuest Timing Analyzer"，打开 "TimeQuest Timing Analyzer"。
2. 选择文件>新建SDC文件。SDC编辑器打开。
3. 在编辑器中键入以下代码。

create\_clock -name "CLOCK\_50" -period 20.000ns [get\_ports {CLOCK\_50}] derive\_pll\_clocks

推导出时钟的不确定度（derive\_clock\_uncertainty

1. 将该文件保存为my\_first\_fpga.sdc（**见图3-34）**。



### 图3-34 默认的SDC

将SDC命名为与顶层文件相同的名称，但扩展名为.SDC，这将导致Quartus II软件默认自动使用这个时序分析文件。如果您使用其他名称，您需要将SDC添加到任务文件列表中。

**第四章**

***编译和验证你的设计***

在创建你的设计之后，你必须对其进行编译。编译将设计转换为可以下载到FPGA的位流。编译的最重要的输出是SRAM对象文件（.sof），你可以用它来为器件编程。该软件还生成其他报告文件，在编译过程中提供关于你的代码的信息。

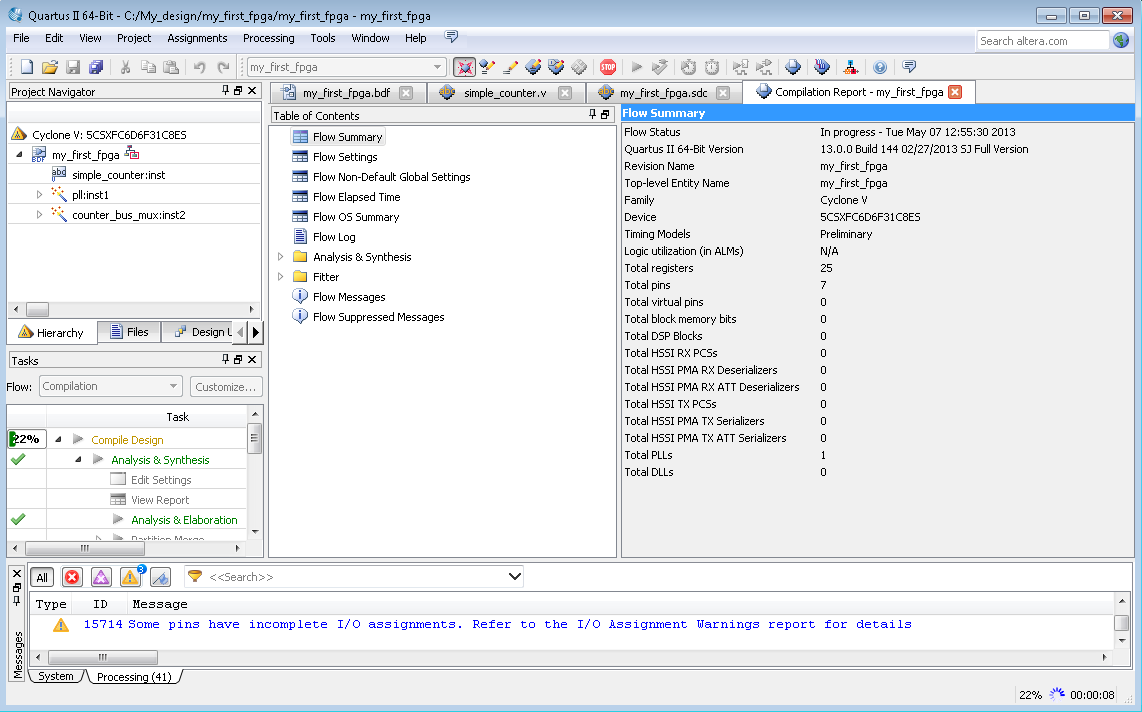
## **汇编你的设计**

如果你想把.SOF存储在存储设备（如闪存或EEPROM）中，你必须首先把SOF转换成专门用于目标存储设备的文件类型。

现在你已经创建了一个完整的Quartus II项目并输入了所有的作业，你可以编译设计了。

在处理菜单中，选择开始编译或点击工具栏上的播放按钮。如果有人要求你保存对BDF的修改，请点击是。

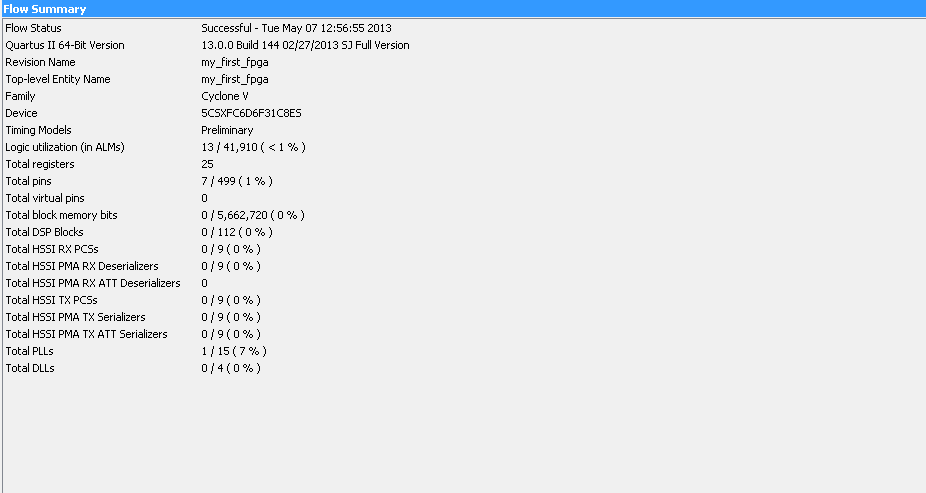
在编译你的设计时，Quartus II软件提供了关于编译的有用信息（[**见图4-1）**](#_bookmark48)。



### 图4-1 项目的编译信息

编译完成后，Quartus II软件会显示一条信息。点击 "确定"，关闭消息框。

Quartus II信息窗口在编译期间显示许多信息。它不应该显示任何关键的警告；它可能会显示一些警告，表明器件的时序信息是初步的，或者用于LED的I/O引脚上的一些参数没有被设置。软件在编译报告标签中提供了编译结果，[**如图4-2**](#_bookmark49)所示[。](#_bookmark49)



### 图4-2 编译报告示例

## **对FPGA器件进行编程**

在编译和验证了你的设计之后，你就可以对开发板上的FPGA进行编程了。您可以使用开发板上的USB-BlasterII电路将您刚刚创建的SOF下载到FPGA中。使用以下步骤设置硬件进行编程。

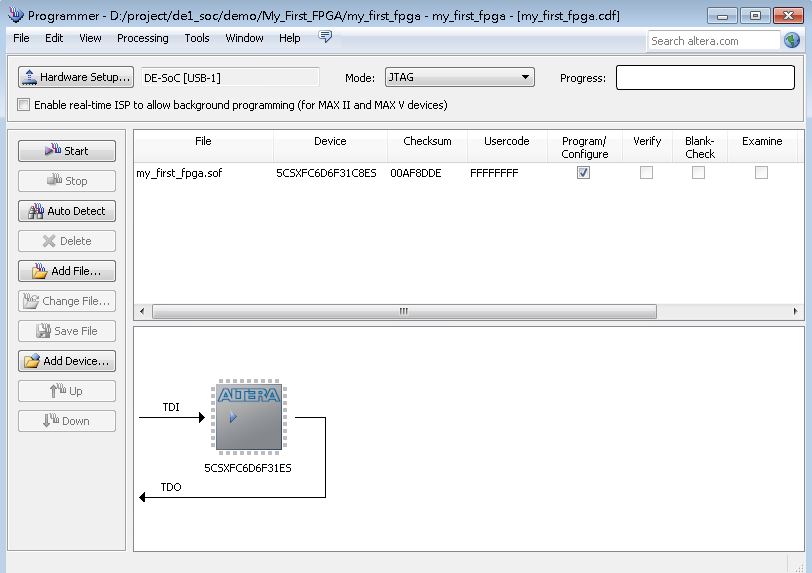
1. 将电源线连接到你的电路板和电源插座上。
2. 对于DE1-SoC板，将USB-BlasterII（包括在你的开发套件中）连接到J13，将USB电缆连接到USB-BlasterII。将USB电缆的另一端连接到主机上。

关于如何连接电缆的详细说明，请参考入门用户指南。

1. 使用开/关开关将DE1-SoC板打开。

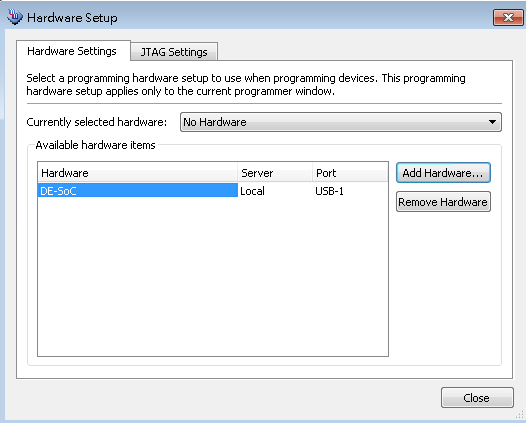
使用以下步骤对FPGA进行编程。

1. 选择工具>编程器。打开 "编程器 "窗口。[**见图4-3**](#_bookmark51)。



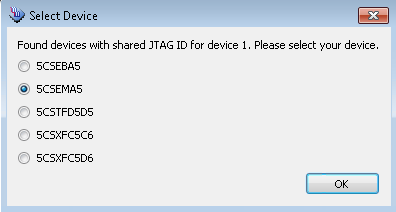
### 图4-3 编程器窗口

1. 点击硬件设置。
2. 如果还没有打开，请打开当前选择的硬件下的USB-BlasterII [USB-1]选项。[**见图4-4**。](#_bookmark52)



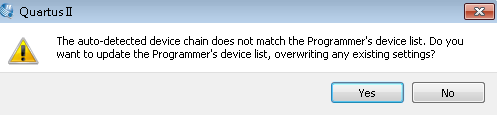
### 图4-4 硬件设置

1. 点击关闭。
2. 点击自动检测来检测JTAG链上的所有设备。
3. 选择5CXC6D6ES，然后点击确定。



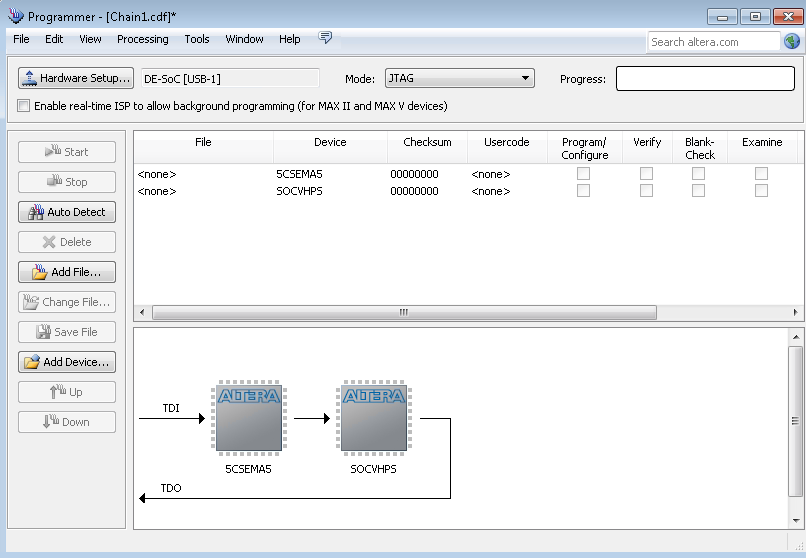
### 图4-5 选择FPGA器件

1. 单击 "是 "以匹配设备列表。



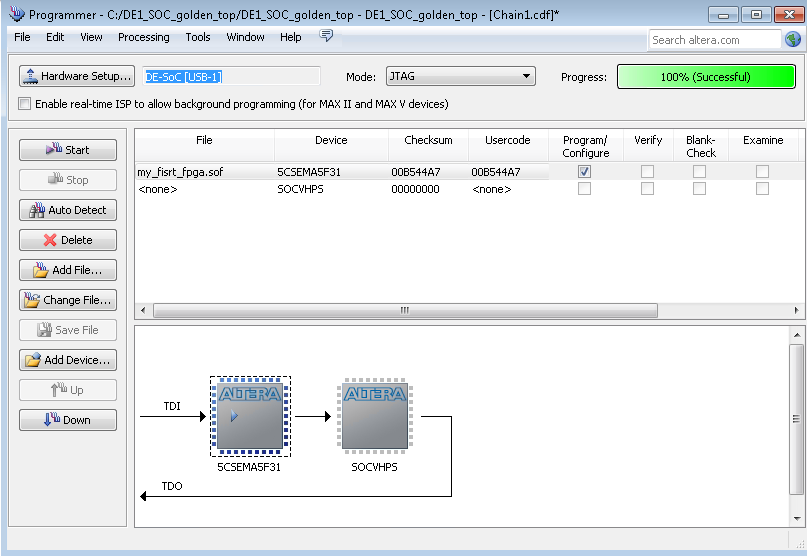
### 图4-6 设备列表不匹配的警告

1. FPGA和HPS都将被列在编程器上。选择FPGA设备并点击添加文件来添加一个.sof文件。



### 图4-7 编程器窗口

1. 从项目目录中选择my\_first\_fpga.sof文件（[**见图4-8**](#_bookmark53)），点击开始，将.sof文件编程到FPGA中。



### 图4-8 下载完成

恭喜你，你已经创建、编译和编程了你的第一个FPGA设计！你的设计是什么？编译后的SRAM对象文件（.sof）被加载到开发板上的FPGA上，设计应该开始运行。

## **验证硬件**

当你在硬件中验证设计时，你观察FPGA硬件设计的运行行为，并确保它的功能是适当的。

通过执行以下步骤验证设计。

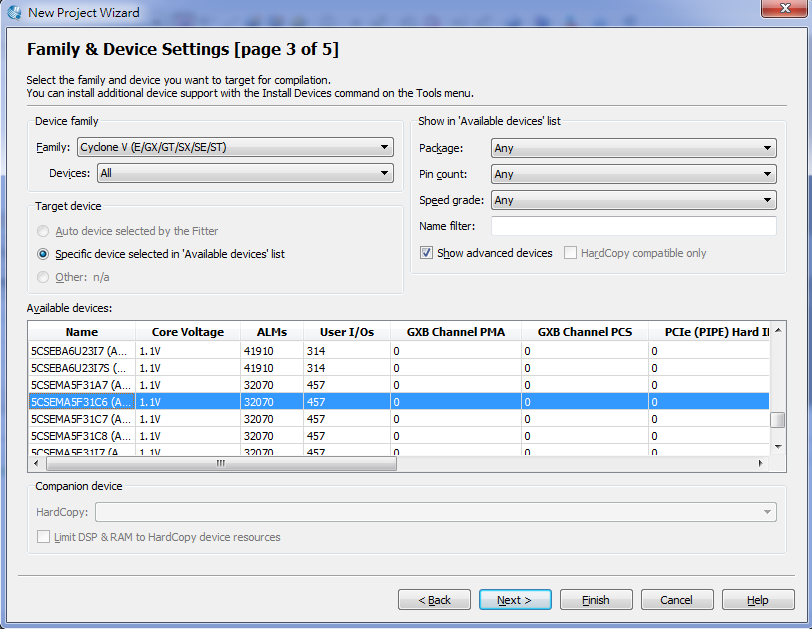
1. 观察一下，四个开发板上的LED似乎在以二进制计数模式缓慢前进，这是由simple\_counter位[26.23]驱动的。

LED是低电平有效，因此，当开始计数时，所有的LED都是打开的（0000状态）。

1. 按住开发板上的键[0]，观察LED的前进速度是否更快。按这个键会导致设计使用更快的前进部分进行复用。

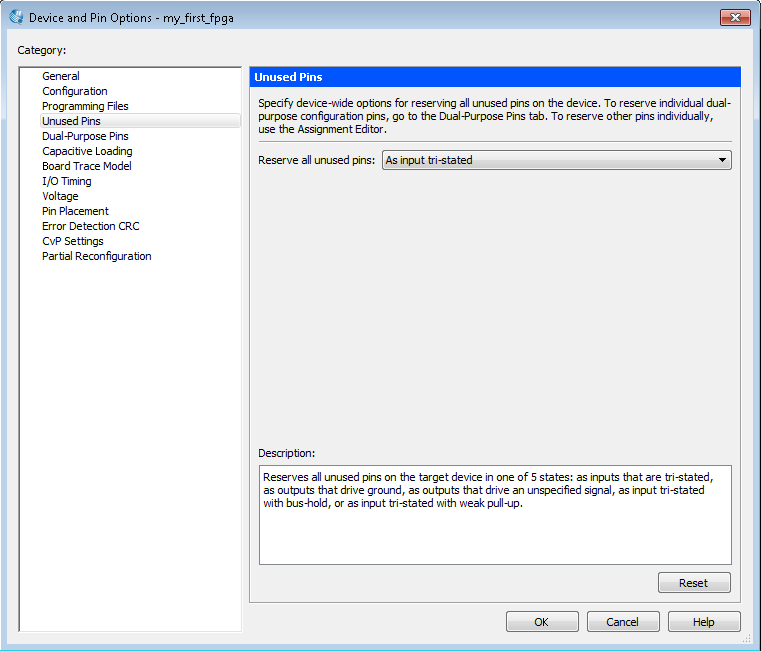
计数器（位[24...21]）。

1. 如果其他LED发出微弱的光，选择分配>设备。点击设备和选项。[**见图4-9**。](#_bookmark55)



### 图4-9 设备和选项

选择未使用的引脚。保留所有未使用的引脚。选择作为输入的三态选项。[**见图4-10**。](#_bookmark56)



### 图4-10 设置未使用的引脚

点击两次确定。

1. 在处理菜单中，选择开始编译。编译后，选择工具>程序员。从项目目录中选择my\_first\_fpga.sof文件。点击开始。这时你可以发现其他的LED是不亮的。

**第五章**

***附录***

## **5.1 总部和分支机构**

* 电话：+886-3-575-0880
* 传真：+886-3-572-6690
* 地址：300-70 新竹市东区公岛五路二段176号9楼
* 电子邮件[：sales@terasic.com](mailto:sales@terasic.com) / [support@terasic.com](mailto:support@terasic.com)