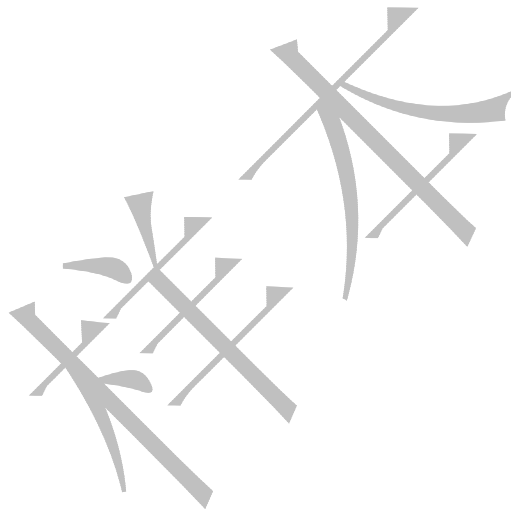


十分钟学会 Xilinx FPGA 设计



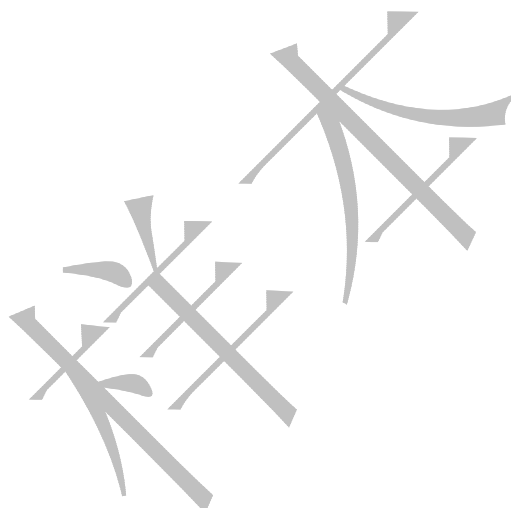
声明

本系列文档、软件的版权为 Digital Power Studio 工作组所有。Digital Power Studio 工作组保留所有权利。欢迎转载，但请保留这段版权声明；请勿用于商业用途。

Copyright (c) 2004, Digital Power Studio. All rights reserved.

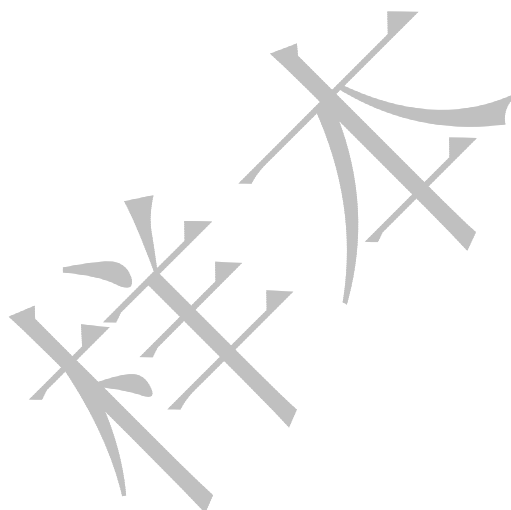
History

Revision	Date	Issuer	Notes
1.0	Mar 3, 2005	Badfish	First Release



目录

概述.....	4
实验目的.....	4
软件准备.....	5
流程介绍.....	5
新建项目：.....	5
编写和导入代码文件.....	8
调用 Modsim 进行仿真.....	11
约束文件.....	13
综合与实现.....	15
下载.....	16



十分钟学会 Xilinx FPGA 设计

概述

这个文档主要帮助大家熟悉利用 ISE 进行 Xilinx 公司 FPGA 代码开发的基本流程。主要是帮助初学者了解和初步掌握 ISE 的使用，不需要 FPGA 的开发基础，所以对每个步骤并不进行深入的讨论。

本文介绍的内容从新建 project 一直下载到硬件观察现象为止，涵盖整个开发过程。考虑到我们的开发一般以 Verilog HDL 或 VHDL 为主，在本文中未介绍原理图输入工具和 IP 核使用工具等。这同时是出于为了使文章脉络更清晰，让大家更快地学会 ISE 的考虑。关于这些专用工具，可以参看 HELP 或者其他文档。

为了有个更直观的理解，我们需要一个实验平台，在本文中，采用的是 Digital Power Studio 工作组 FPGA SPARK1.1 综合开发平台的 Xilinx 标准型开发系统。如图 1 所示（图中为 Altera 子板，只需改成 Xilinx 的即可）。在该开发系统中，所采用的芯片是 SPARTAN II 系列的 XC2S200。可以参看 <http://www.fpga.com.cn/solution/fpgaspark.htm> 中对该系统的介绍。



图 1 FPGA SPARK1.1 外观图

实验目的

在如上系统中右边有一个 128×64 带背光的点阵型液晶。我们的实验就是用 Verilog HDL 编写一段代码驱动液晶显示一个流氓兔。其效果如图 2 所示：

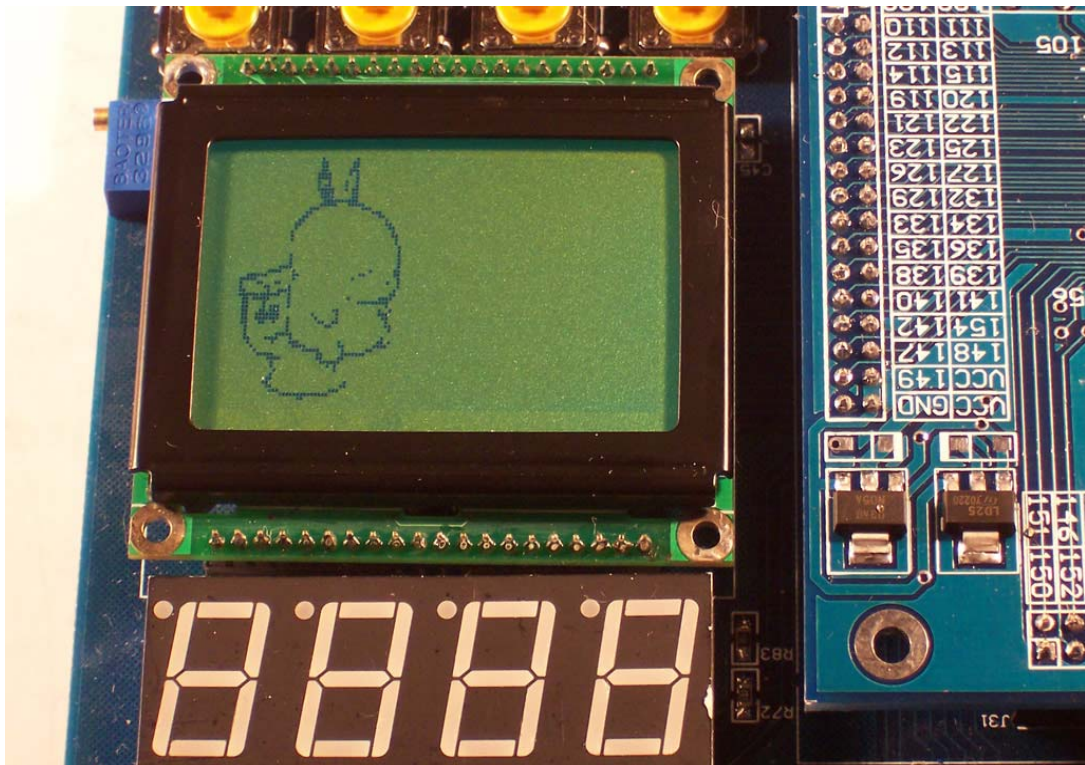


图 2 点阵液晶上显示流氓兔的效果图

软件准备

本文介绍的是 Xilinx 公司的 ISE 开发环境。现在最新的版本是 ISE7.1，其界面与 ISE5.x/ISE6.x 一致。为了更好地进行仿真，还需要安装 Modsim。但是 ISE 本身可以进行仿真，也可以不安装 Modsim。

流程介绍

新建项目：

在开始→程序→Xilinx ISE 中找到 Project Navigator，点击打开。也可以通过双击桌面上 Project Navigator 的快捷方式打开。ISE 开发环境如图 3 所示：

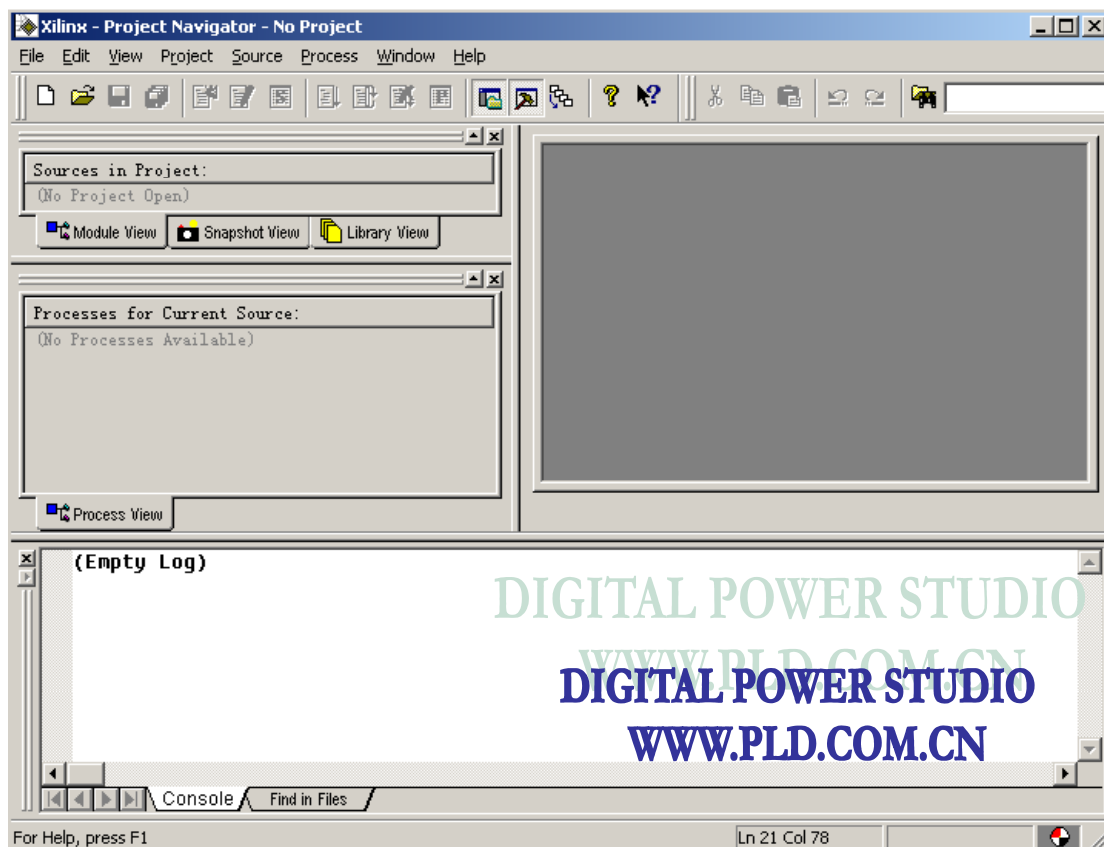


图 3 ISE 开发环境

图 3 中选择 File→New Project, 进入图 4;

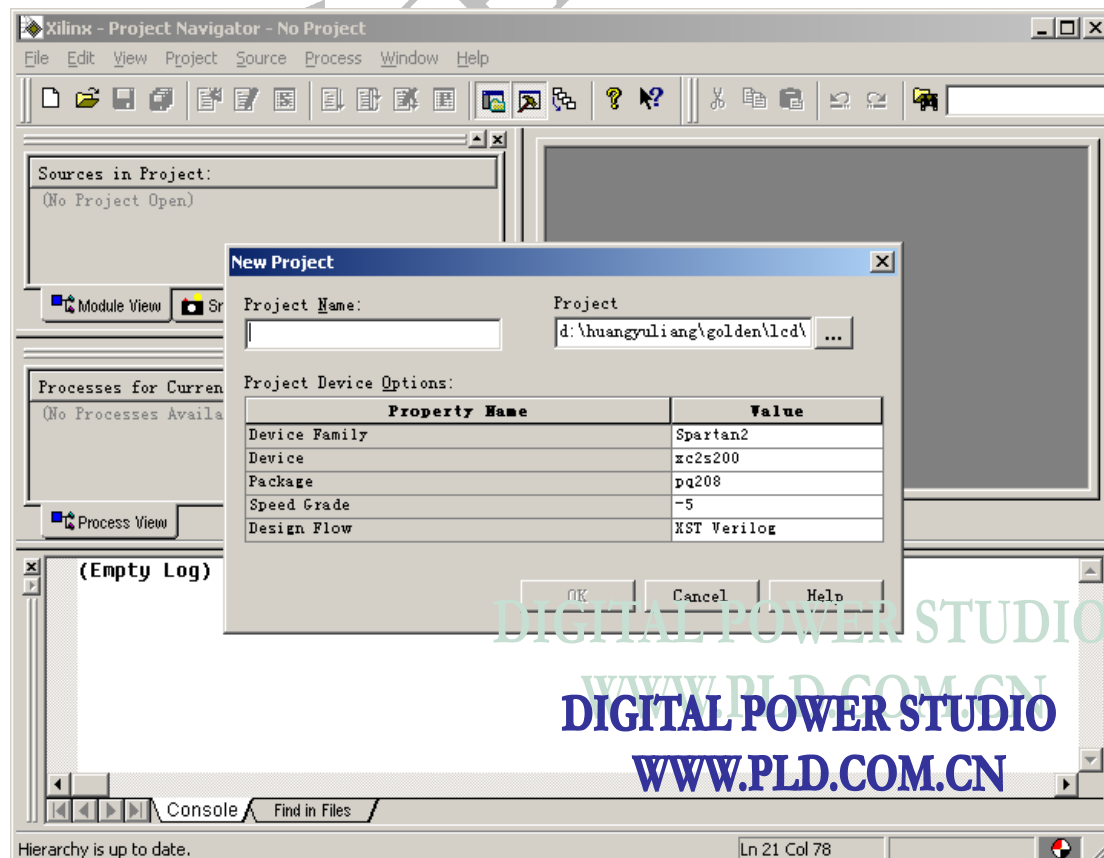


图 4 新建 project 界面

在 Project Name 栏输入项目名称, Project 中填入该项目的目录, Project Device Options 中, Device Family 为所用芯片所在系列, Device 为所用芯片, Package 为所用芯片的封装形式 (每个芯片可能有不同封装的几种, 如 XC2S200 有 3 种封装的), Speed Grade 为速度等级, Design Flow 为所选用综合方式 (综合工具和代码语言)。

由于采用的是 FPGA SPARK1.1, 采用如图 5 的设置。

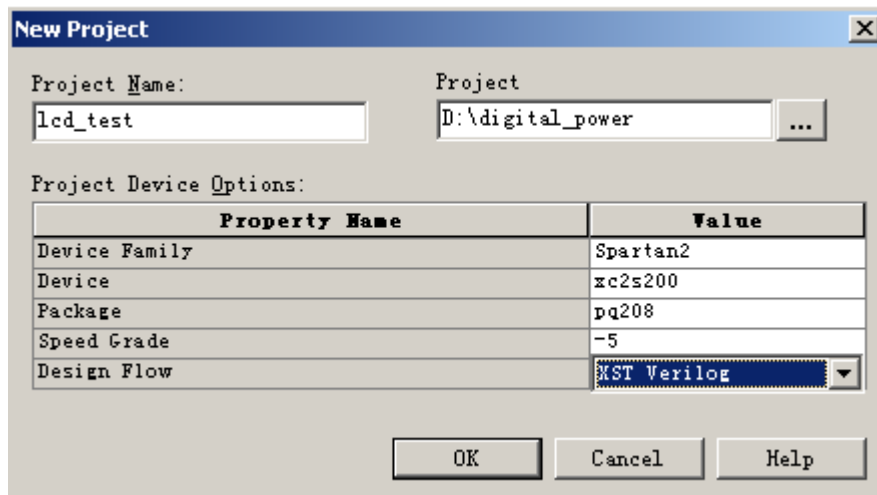


图 5 新建项目的信息输入

其中 XST Verilog 表示采用 ISE 自带的综合器, 代码语言为 Verilog HDL, ISE 支持其他综合工具的调用, 也支持标准 edif 文件的输入。XST 是 ISE 自带的综

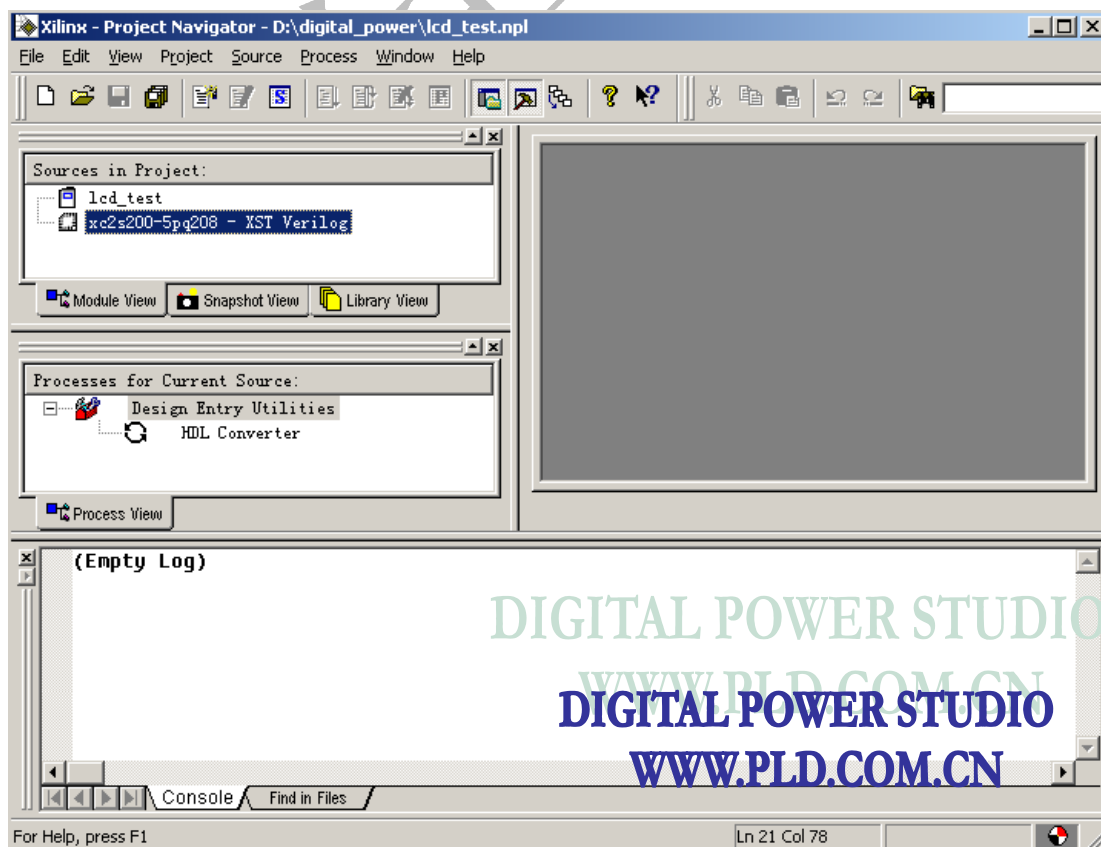


图 6 新建项目最终效果图

合工具，由于 Xilinx 对自身的硬件了解程度比任何第三方软件开发商都更深，故 XST 的综合性能有自身独特的优点。

点击 ok，新建项目完成。可以在相应目录下看到生成的文件。效果如图 6。如果要更改硬件的设置，可以双击图 6 中模块视窗内蓝色选中的所示对应选单，即可进入 project properties 界面重新进行设置。

编写和导入代码文件

如图 7 中所示，在 Project 菜单下，选择 New Source 建立新的代码文件。弹出的对话框如图 8 所示。

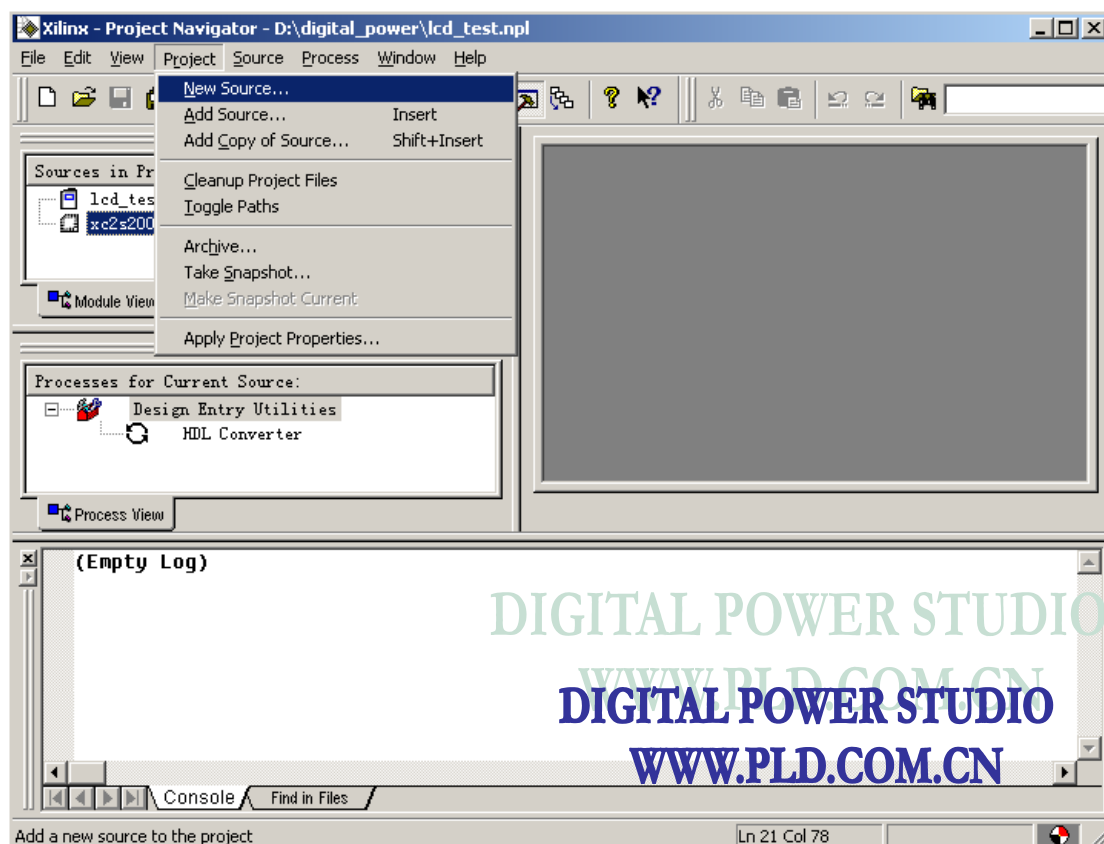


图 7 Project 菜单

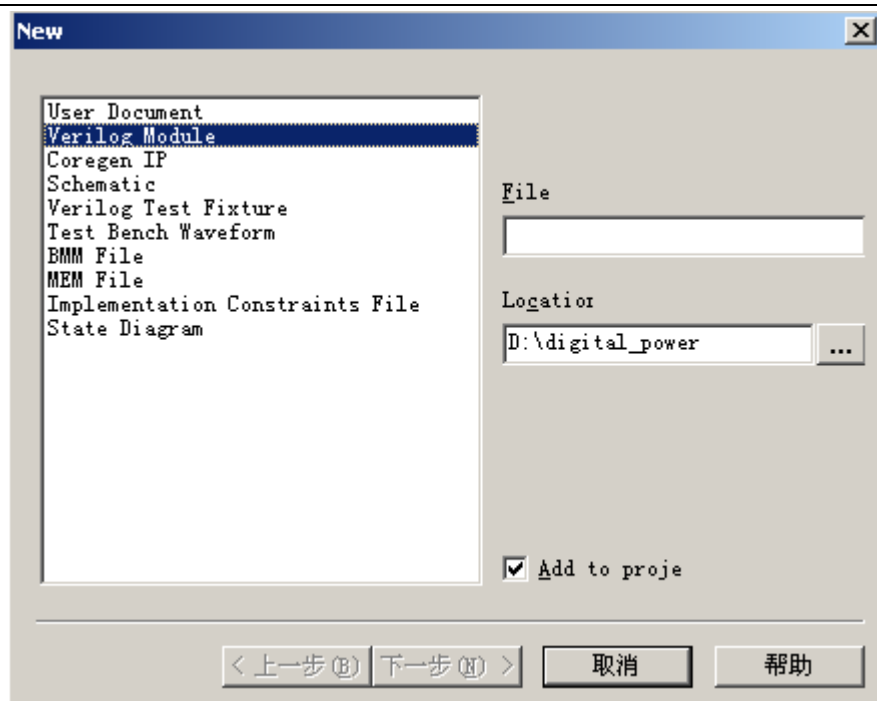


图 8 新建代码文件对话框

在 File 中输入文件名称，Location 最好选用和 project 文件所在目录同一个目录。将 Add to project 选单选中，该文件将自动被加入当前 project 中。在左边窗口中选择建立的文件类型，如图 8 选择为 Verilog Module。点击下一步，如图 9 所示。

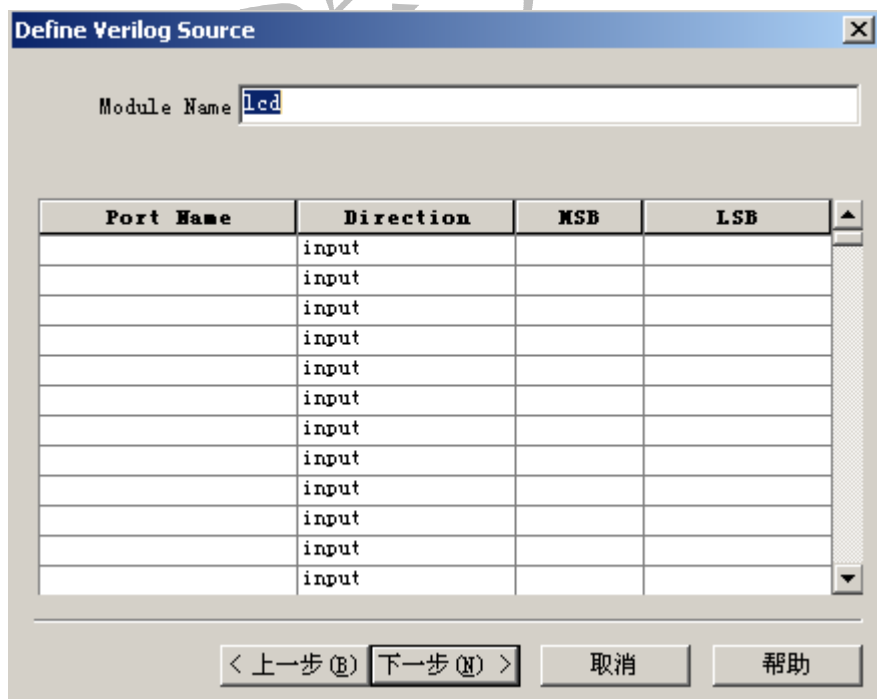


图 9 Verilog 代码文件定义对话框

在图 9 中，可以定义代码文件的输入输出端口，可以不进行定义，直接点击下一步。如果进行了定义的话，生成的代码文本文件会自动加上定义好的输入输出端口。点击下一步，进入下一对话框后点击完成，即可。如图 10 所示。在图 10 中，可见左面 Module View 中 project 中已经多处了刚才建立的 lcd 模块，右

边打开的 lcd 文件界面中，可以进行代码的编写。

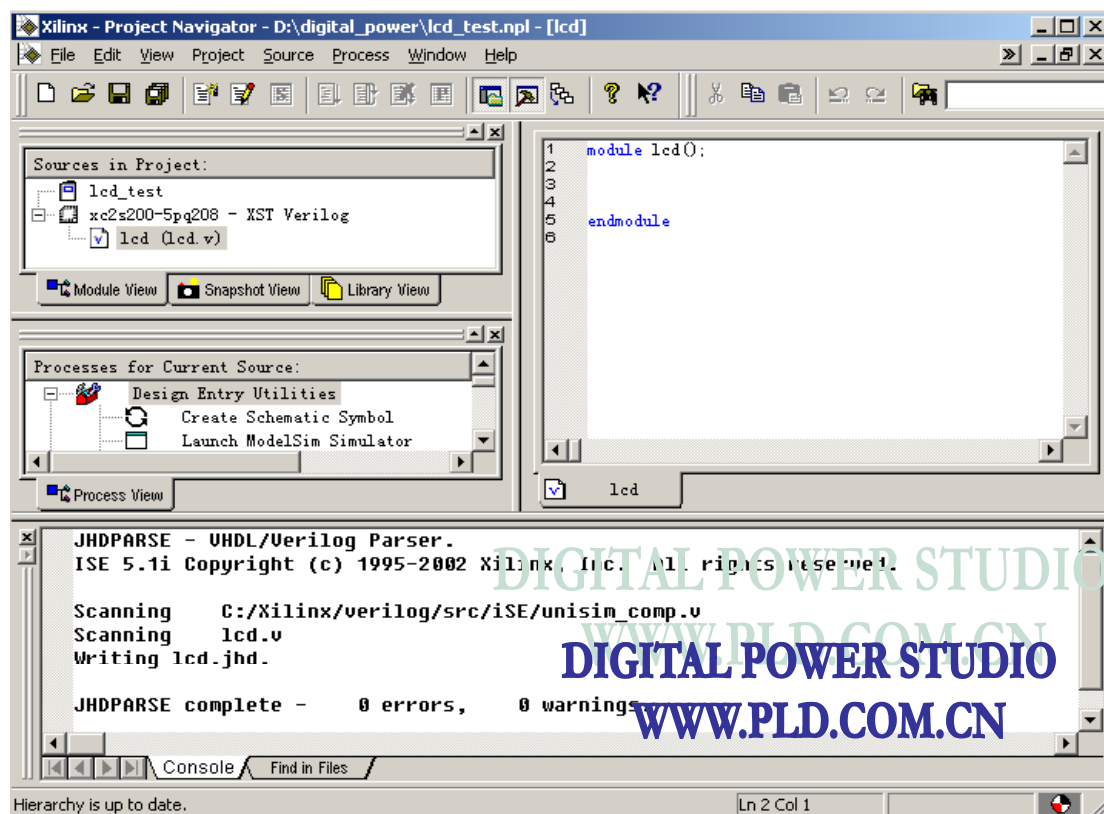


图 10 代码文件生成后界面

前面介绍的是编写代码，如果已经用其它工具编写好了代码，或者有现成的代码，则不需要再重新编写了，下面介绍如何将已有的 verilog 文件导入到 project 中。

在图 7 中不要选择 New Source，而选择 Add Source。在弹出的窗口中找到已有代码文件（可同时加多个文件），并点击“打开”即可。注意：Add Source 是将当前选择的代码加入到当前 project 中；而 Add Copy of Source 是将代码文件复制到当前 project 所在目录中，并将复制后的文件加到 project 中。建议使用后者。

添加后结果如图 11 所示。（在本文例子中，加入的是 Digital Power 提供的液晶程序 lcd.v。）可以看到在 Module View 窗口中将显示出 verilog 代码中所有 Module，并显示出其层次结构以及其所在的文件。

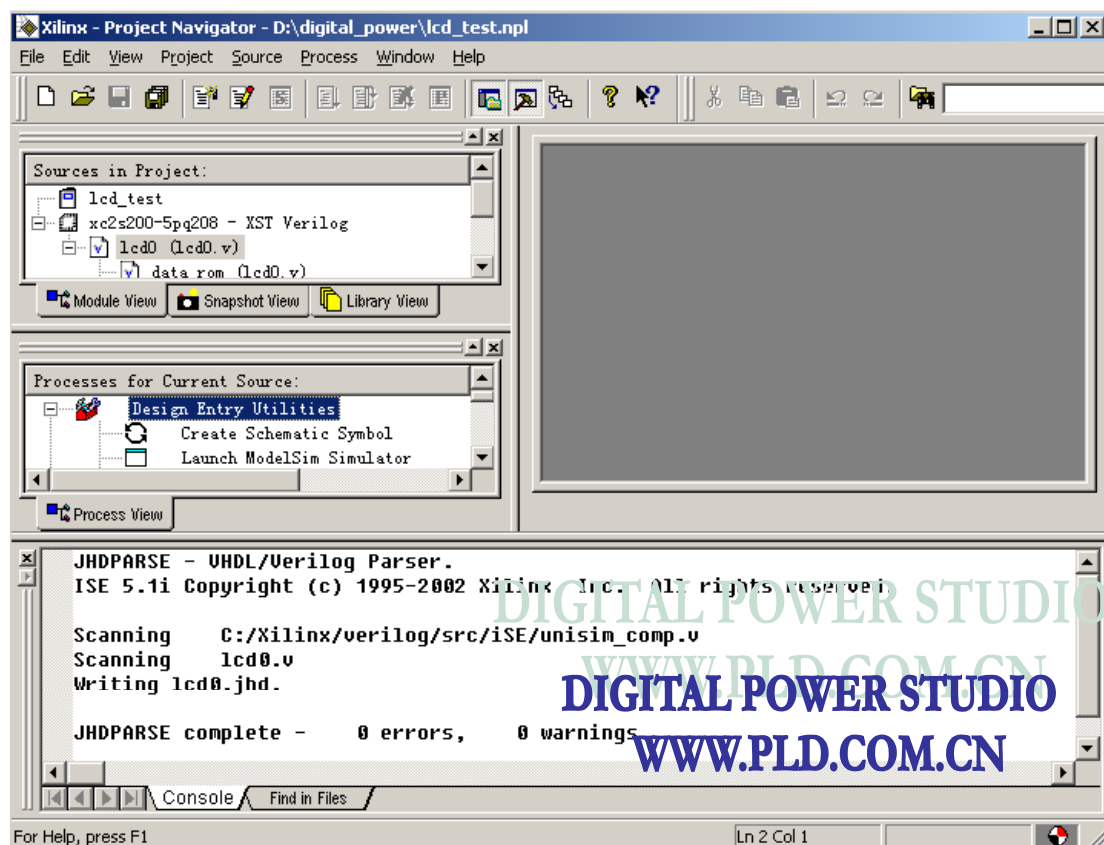


图 11 Add Source 结果图

调用 Modsim 进行仿真

由于 modsim 的性能非常优秀，以及它的普及程度广，在本文中，介绍如何调用 modsim。

首先，在 edit 菜单中选择 Preferences 选项，弹出如图 12 所示对话框。在 Partner Tools 中的 Model Tech Simulator 输入框中填入您所安装的 Modsim 应用程序文件的路径。若您所装的 Modsim 版本为专门为 Xilinx 编写的 Modsim xilinx edition 版本，可以省略该步骤。

经过设置以后，回到如图 13 所示界面。

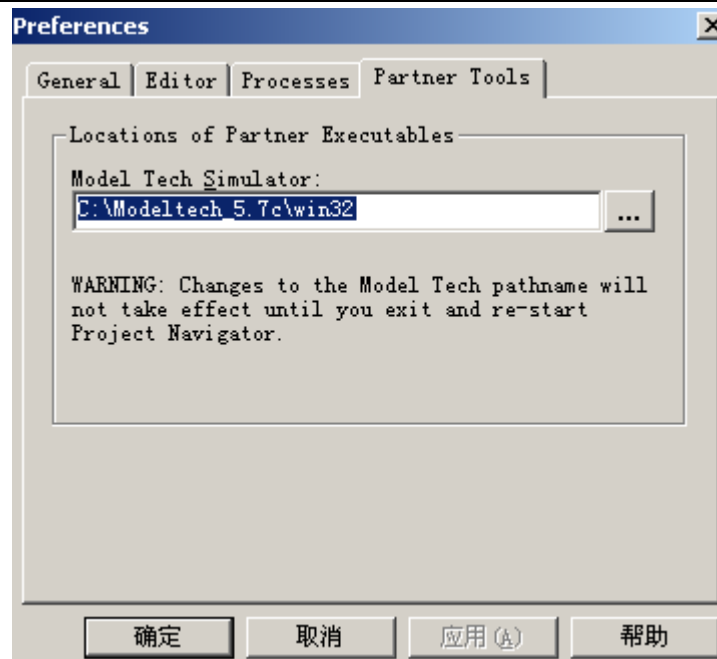


图 12 Preferences 对话框

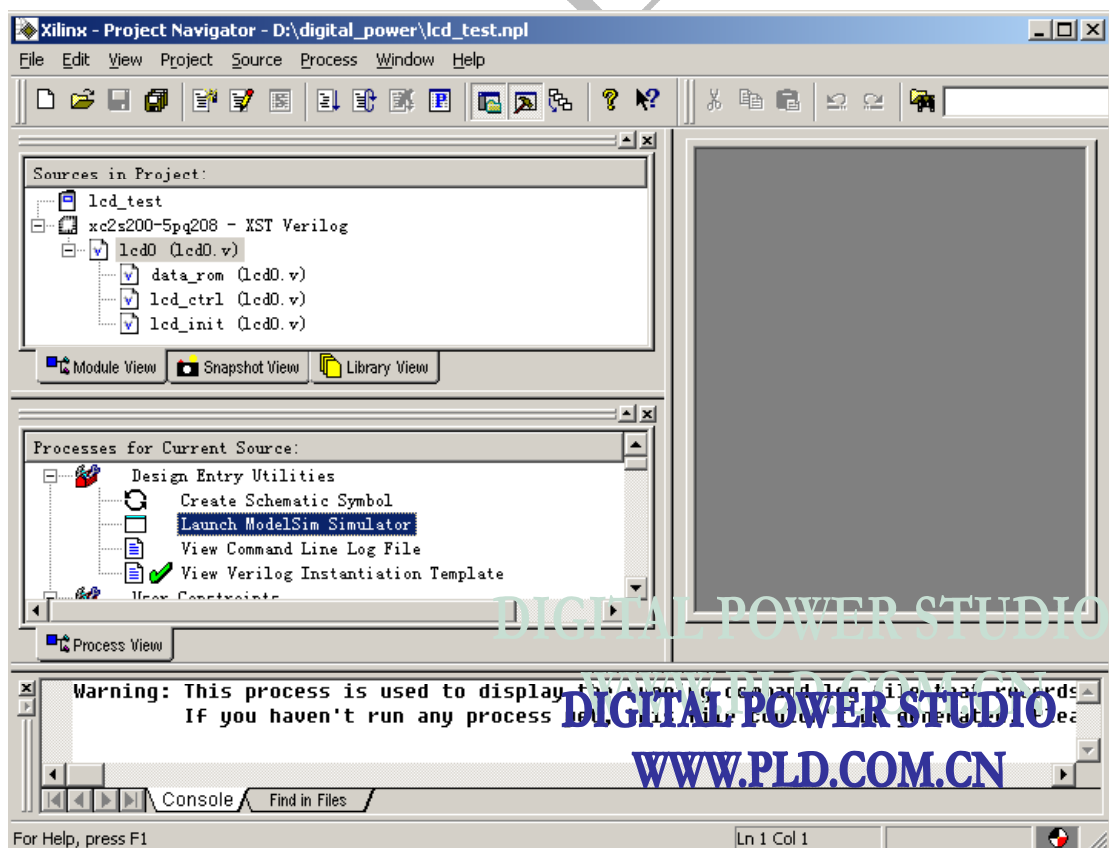


图 13 调用 Modsim

图 13 中 Module View 窗口中选中代码的最顶层模块。而后如图 13 蓝色部位所示，在 Process View 窗口中的 Launch Modelsim Simulator，将自动打开 Modsim，并且已将当前 project 的相关信息加载到了 Modsim 中。如图 14 所示。要注意的是，后仿真也是如此调用 Modsim。关于 Modsim 的使用请参考相关文档。

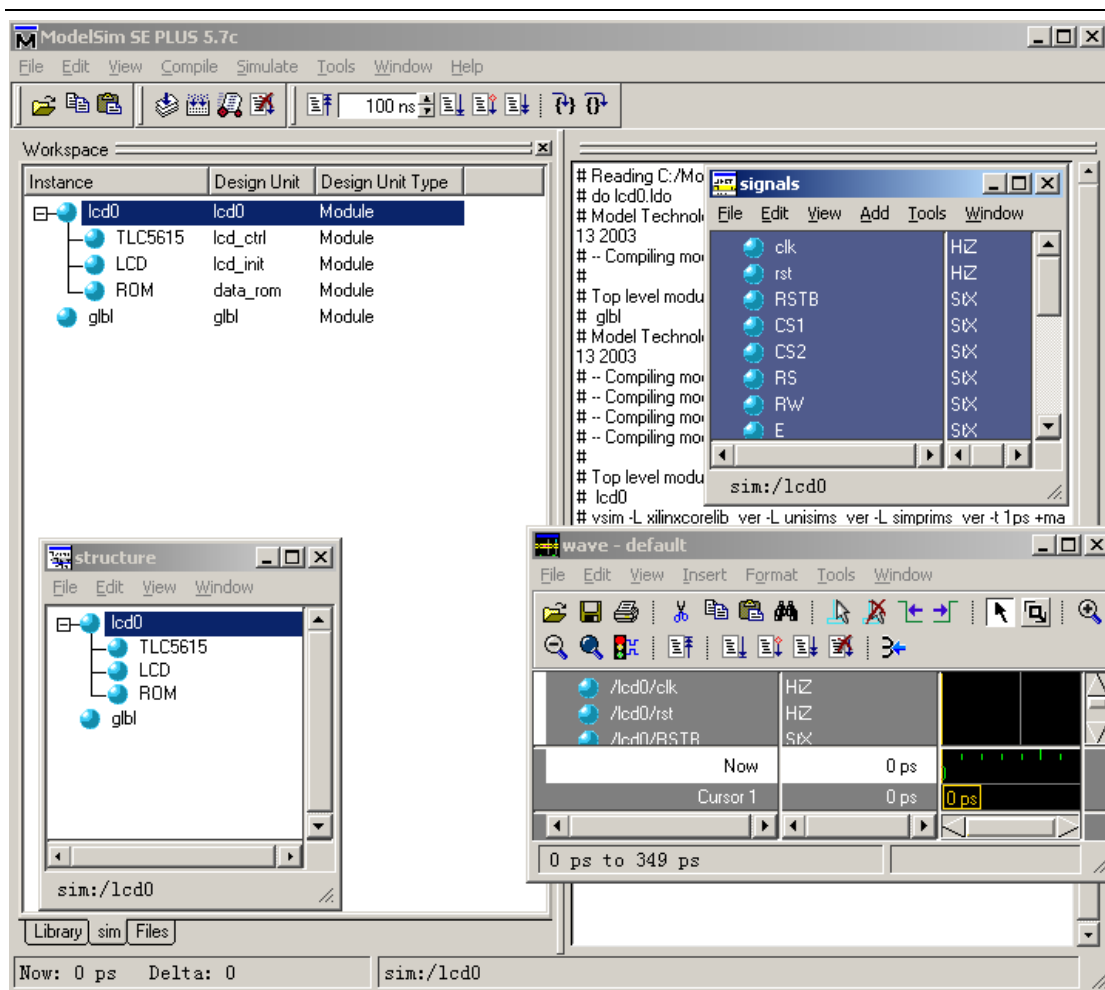


图 14 调用的 Modsim

如果需要仿真的不是最顶层模块，则在 Module View 窗口中选中相应的代码文件，用同样的步骤调用仿真。

约束文件

ISE 是一款基于约束的开发工具。在进行高速数字电路设计的时候，需要在综合、映射、布线时附加约束条件，以便控制综合、映射、布线过程，使设计满运行速度，使用资源和引脚位置等要求。

约束包括时序约束、分组约束和区域约束。在本文中，将介绍最常用的 Constraints Editor 约束编辑器，以及时序约束和引脚绑定（该两者最为基本）。

首先，像第二步编写代码文件中一样建立约束文件（.ucf 文件），在本文中，如图 15 所示，可以看到 lcd_c.ucf 文件即为一个约束文件。新建时，唯一的不同是在图 8 中左面对话框中应选择 Implementation Constraints File。同样地，如果已经有现成的约束文件，也可以通过和代码文件相同的方法导入。

如图 15 所示，在 Module View 窗口中选中代码的最顶层模块，而后在 Process View 窗口中双击 Create Timing Constraints 或者 Create Area Constraints，将自动打开 Constraints Editor 约束编辑器。

需要注意的是，双击 Assign Package Pins 将打开引脚与区域约束编辑器 PACE，由于 Constraints Editor 约束编辑器完全可以完成引脚绑定的功能，对于 PACE 的使用，本文不再进行详细讨论。

Constraints Editor 如图 16 所示。

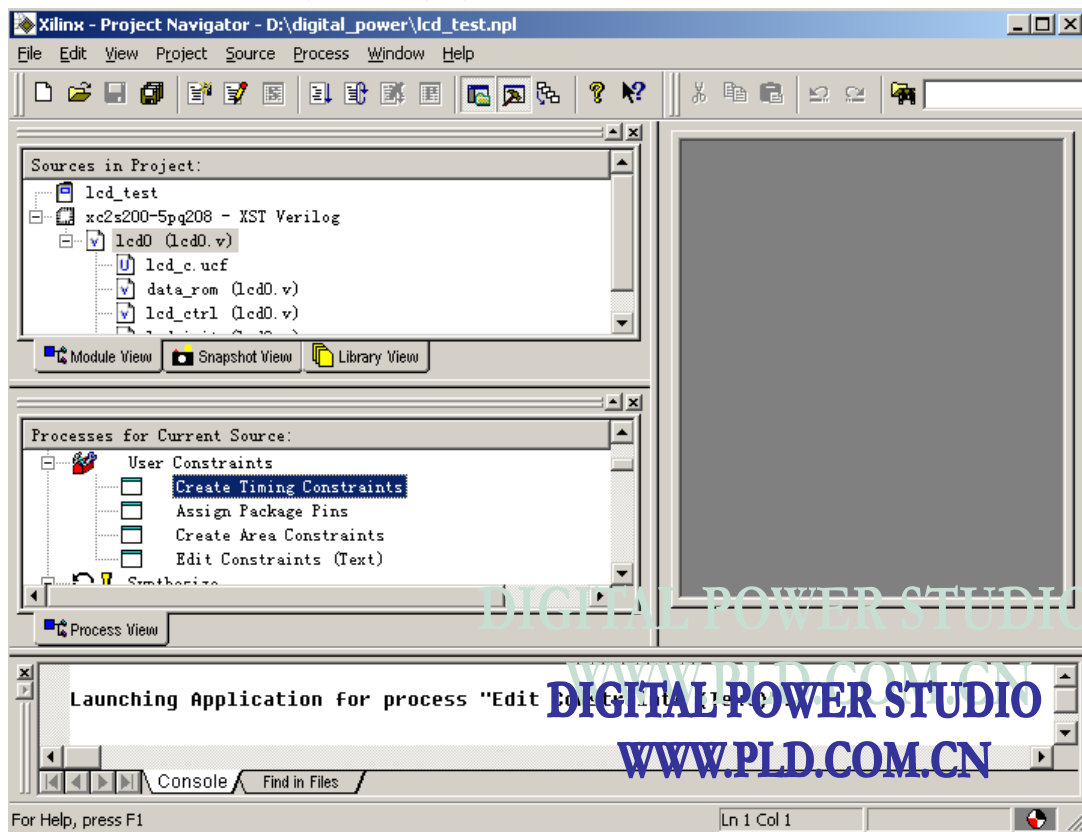


图 15 在 project navigator 中调用 Constraints Editor

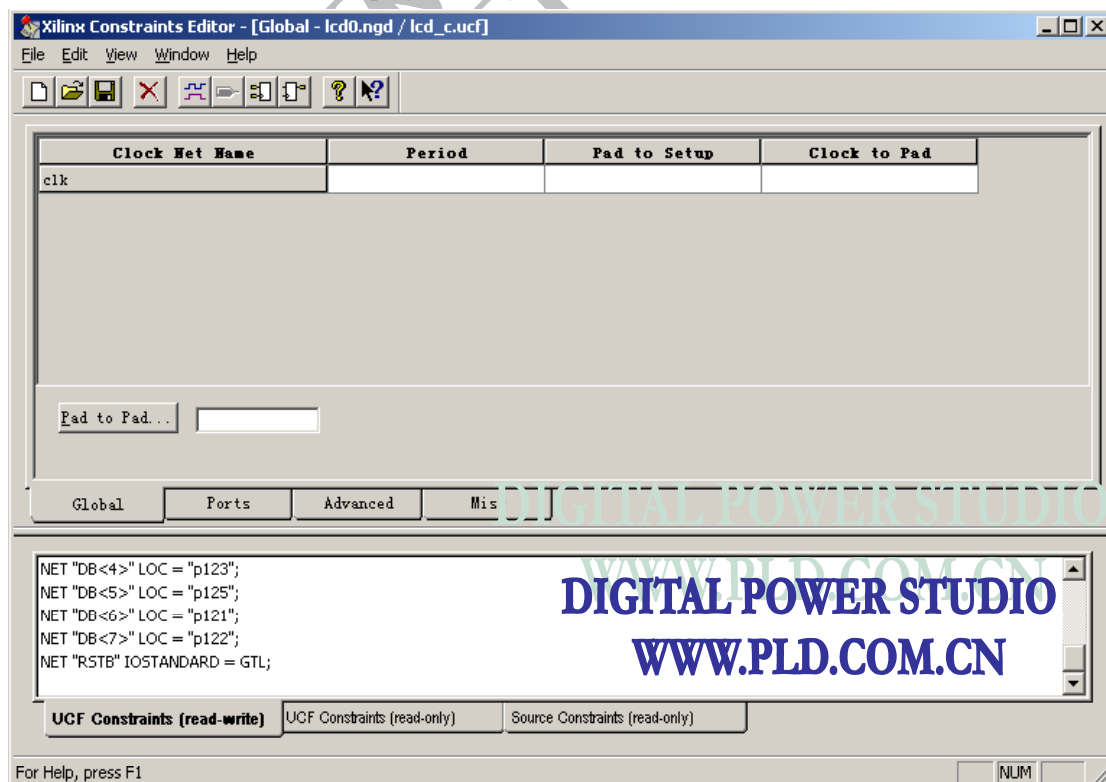


图 16 Constraints Editor 界面

如图 16 所示，共有四类约束界面，Global 为附加全局约束，Ports 为附加端口约束，Advanced 为附加分组约束和时序约束，Misc 为附加专用约束。

常用的是 Ports，其中可以进行引脚绑定，如图 17 所示。双击对应信号的 Location 框，可以进行输入。在引脚号前必须加上字母 p。

在这次 LCD 实验中，可以直接导入 digital power 提供的 lcd_c.ucf 文件，也可以自己在 Constraints Editor 中如图 17 中所示一样进行设置。

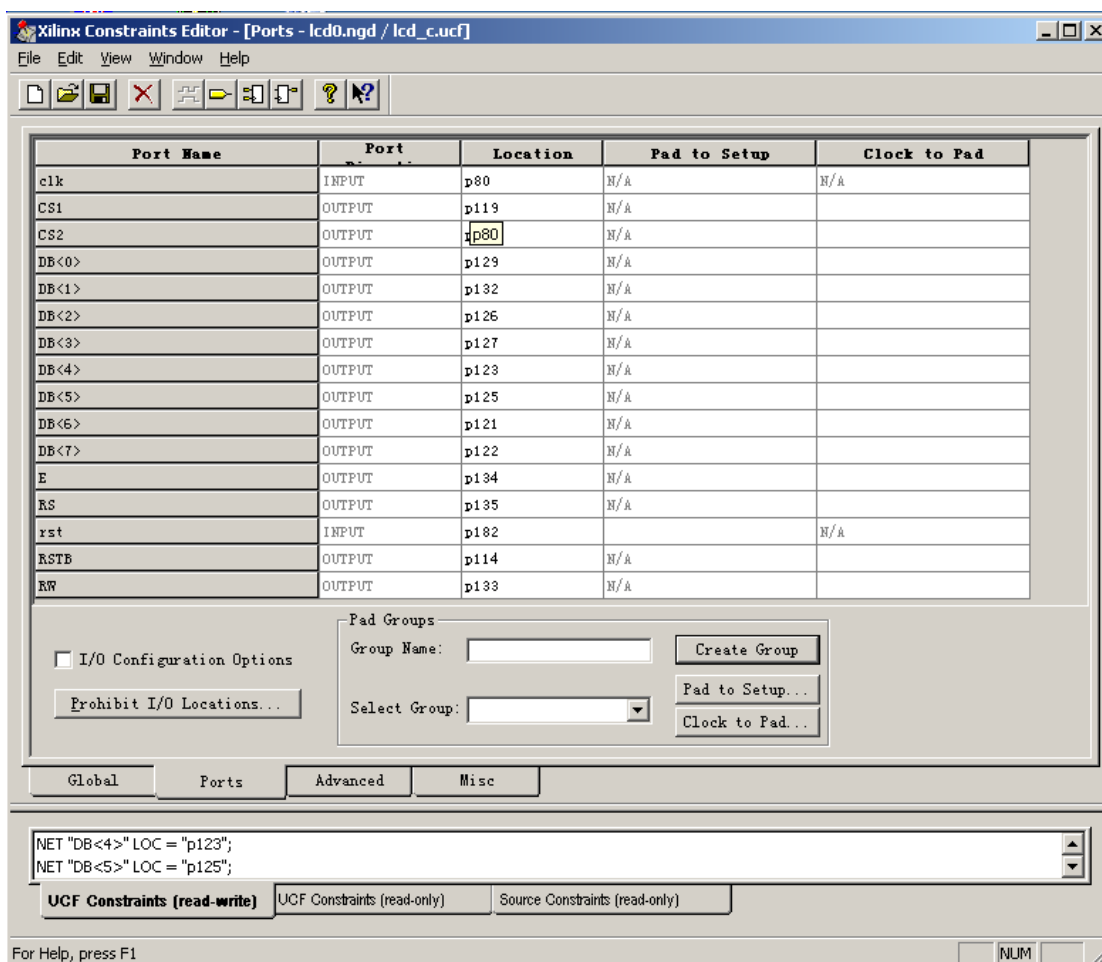


图 17 附加引脚约束

通过以上设置，就完成了约束，在这里附加提到一点就是，在 Constraints Editor 中，下方的对话框将显示 ucf 文件的内容。

约束文件也可以通过文本输入的方式完成，在图 15 中双击 Edit Constraints (Text) 即可。

综合与实现

在进行完约束以后，就可以进行综合与约束了。综合与实现的工作将由 ISE 自动完成，用户通过约束文件和对 properties 选项控制实现对整个过程的控制。整个过程将产生映射、布线等一系列时序报告以及 RTL schematic View，还可以

利用 XPower 进行功耗分析，便于用户对综合与实现的结果有个详细的了解。
综合与实现如图 18 所示。

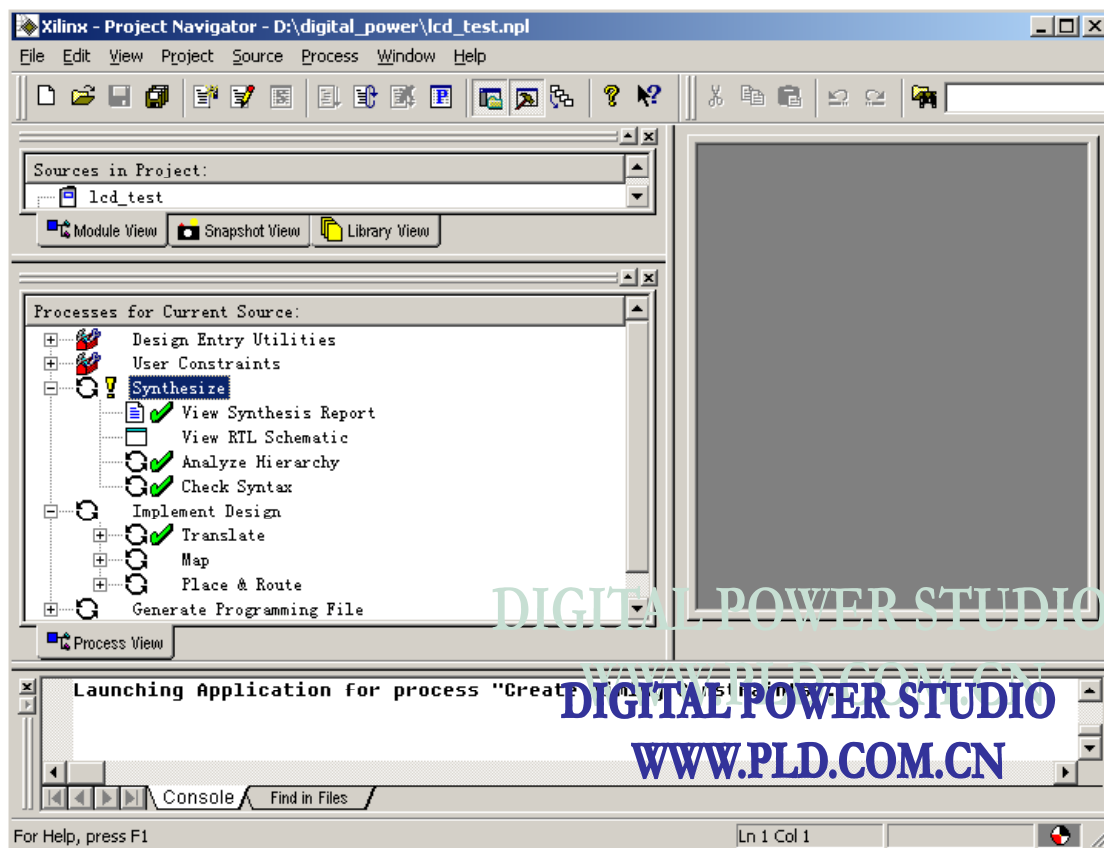


图 18 实现与综合

在 process view 中，Synthesize 为综合过程，Implement Design 为实现过程。在其上点击右键选择 properties 选项，将进入 properties 对话框，可以对过程要求进行控制。一般直接点击右键选择 run all 即可。

下载

最后就只剩下下载工作了。在下载之前，必须先将硬件准备好。
首先，如图 19 所示，连接好电源线。

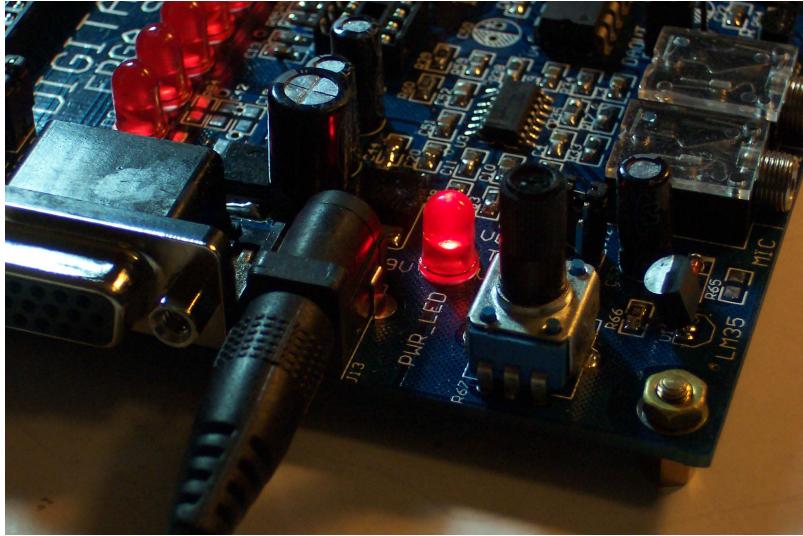


图 19 连接电源线

其次，如图 20 所示，连接好下载线。

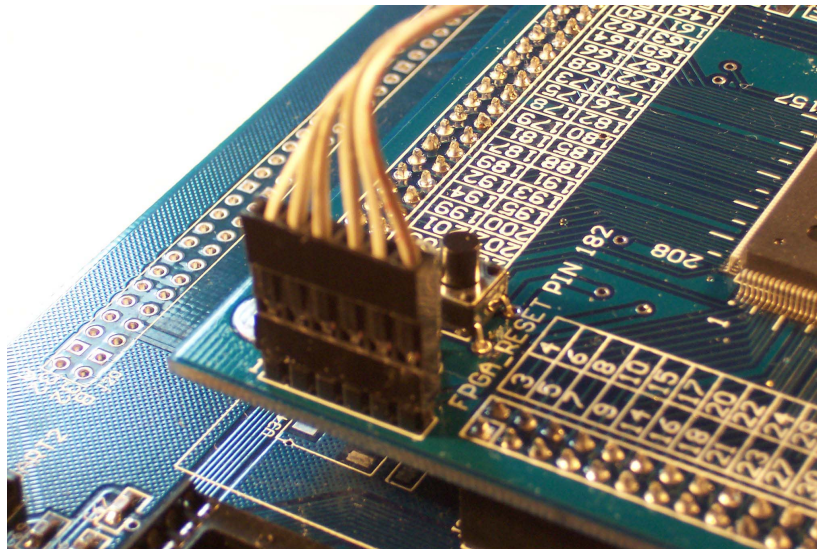


图 20 连接下载线

在硬件准备好后，如图 21 所示，双击 Configure Device (iMPACT)，打开 ISE 中集成的芯片下载工具 iMPACT。

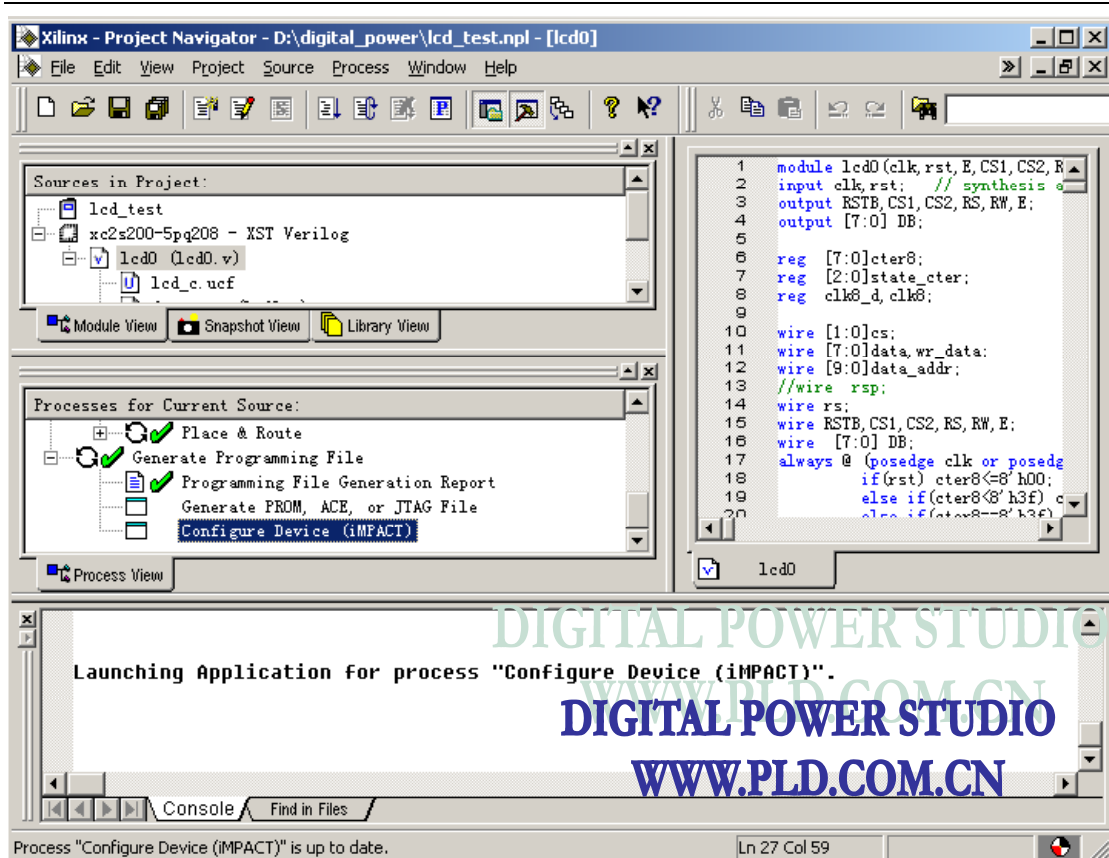


图21 打开 iMPACT

iMPACT 如图 22 所示。

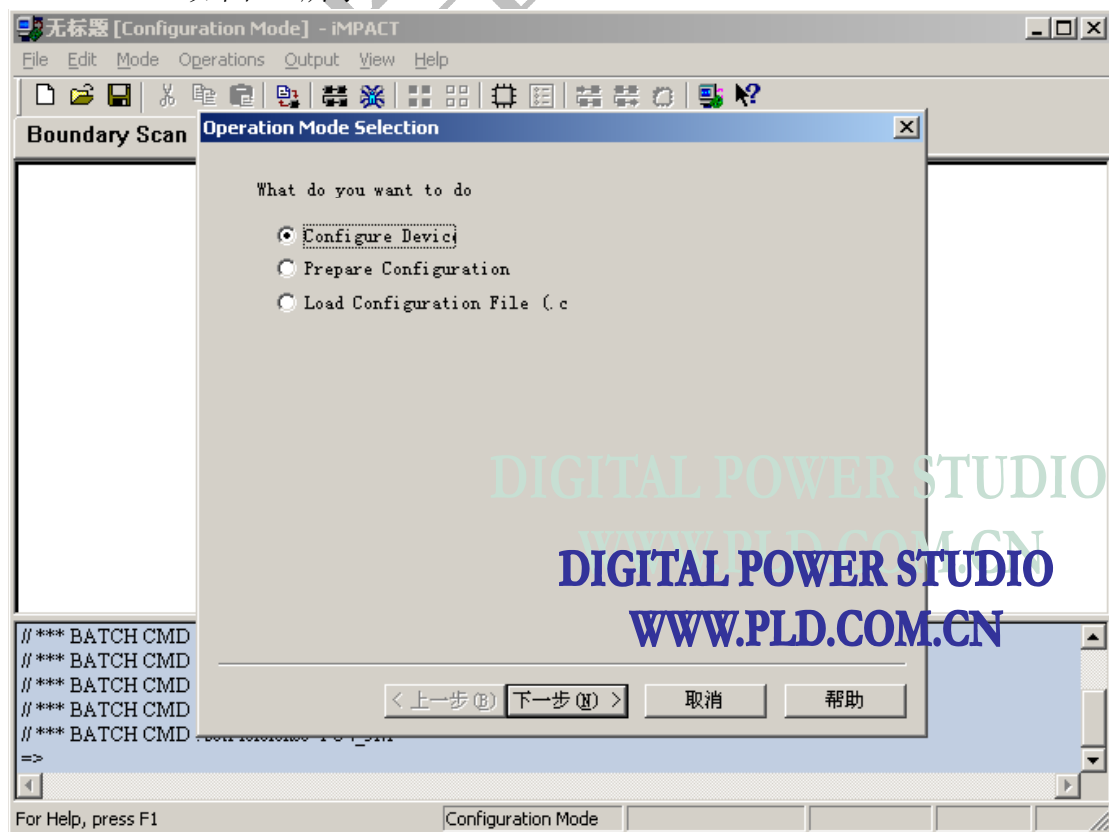


图 22 iMPACT

在图 22 中，点击下一步。

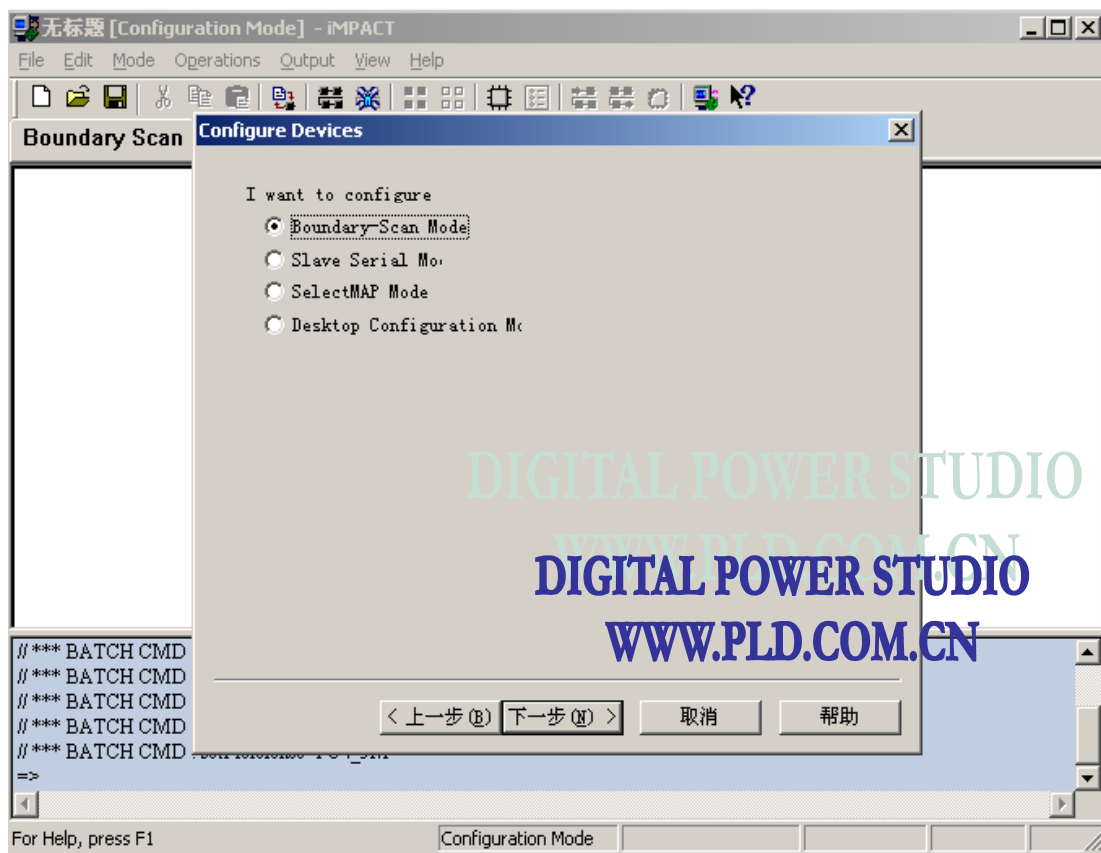


图 23 选择配置 FPGA 的方式

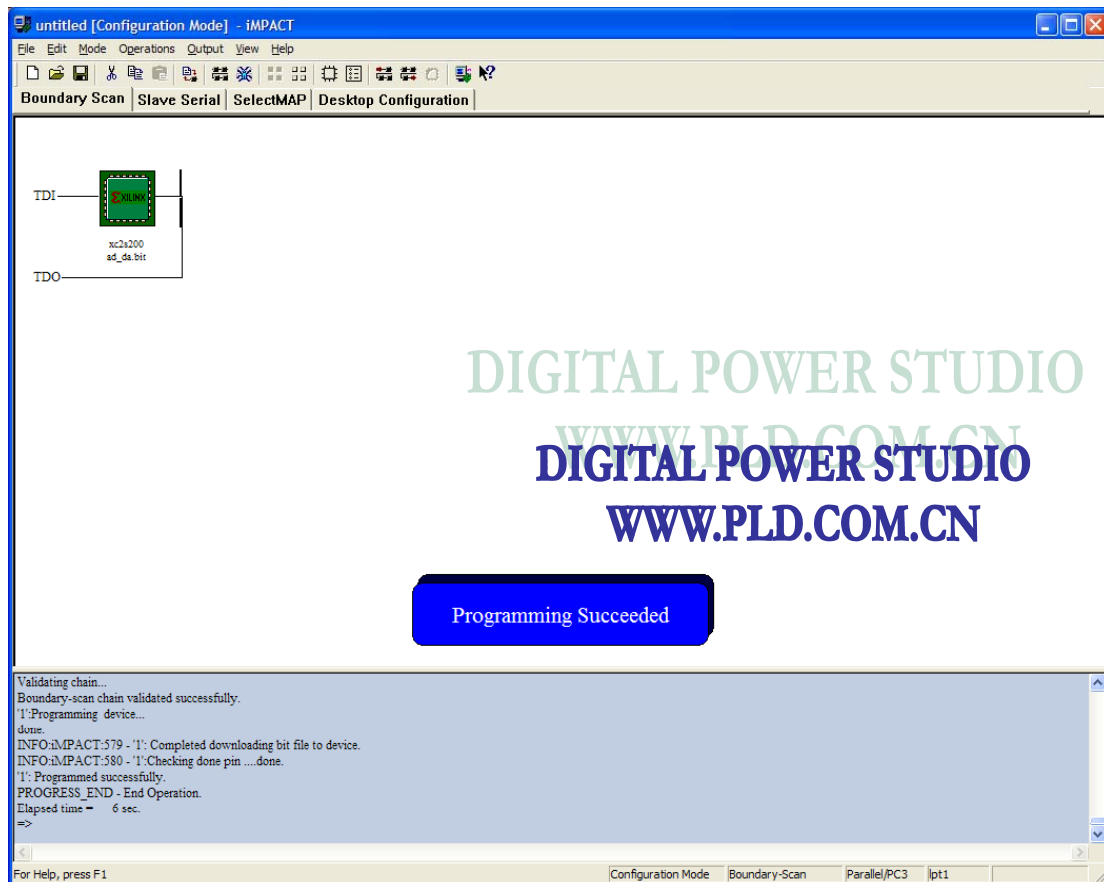
如图 23 所示，是选择配置 FPGA 的方式，选择第一项边界扫描（JTAG）方式。点击下一步。进入下一界面后，点击完成。

若硬件连接正确，将进入图 24 所示界面，XC2S200 芯片将被自动搜索到，显示出来。若硬件连接有问题，可以在正确改接硬件电路以后，在空白处点击右键选择 Initialize Chain 略过以上一些步骤直接开始搜索芯片。若连接有问题，将弹出对话框选择连接端口，一般采用默认的 LPT 端口（并口）即可。

双击图中 XC2S200 的图标，在弹出的窗口中选择你所需要下载的 bit 文件，点击 ok 即可。

在 XC2S200 上点击右键选择 PROGRAMME，并在弹出的对话框中选择 programme 将开始下载。

下载成功后如图所示。



这时你就可以在液晶上看到如图 2 所示的流氓兔效果了。