

“你的 Nexys 4DDR 第一个入门实验” 官方指导手册



第一章 : Nexys 4 DDR 硬件

Nexys4-DDR采用了Xilinx Artix-7 FPGA芯片, 它是一款简单易用的数字电路开发平台, 可以支持在课堂环境中来设计一些行业应用。大规模、高容量的FPGA, 海量的外部存储, 各种USB、以太网、以及其它接口, 这些让Nexys4-DDR 能够满足从入门级组合逻辑电路到强大的嵌入式系统的设计。同时, 板上集成的加速度、温度传感器, MEMs数字麦克风, 扬声器放大器以及大量的I/O设备, 让Nexys4-DDR不需要增添额外组件而用于各种各样的设计。

主要规格/特殊功能

Nexys4的采用了Xilinx Artix-7 100T FPGA芯片。Artix-7 FPGA针对高性能逻辑进行过优化, 它相比之前的FPGA提供了更大的容量, 性能更强且资源更多。对于Nexys4而言, 其规模相当于上一代Nexys3的6倍, 经典的Spartan 3E Starter Kit的9倍。

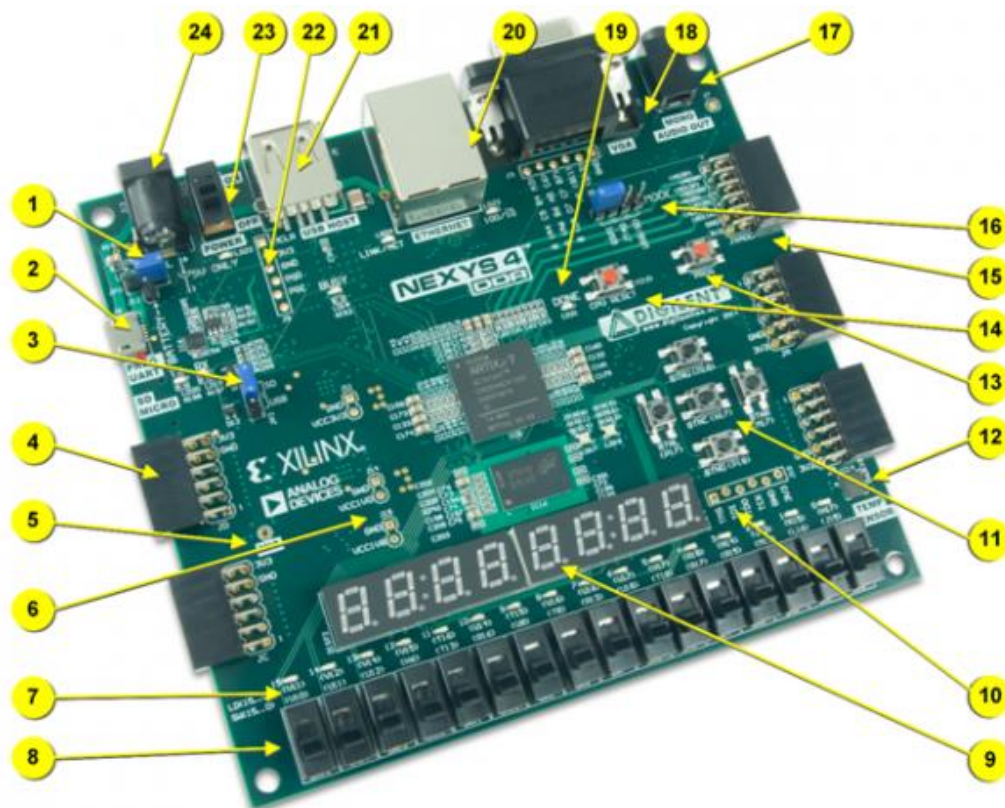
关键特性

Nexys4的采用了Xilinx Artix-7 100T FPGA芯片。Artix-7 FPGA针对高性能逻辑进行过优化, 它相比之前的FPGA提供了更大的容量, 性能更强且资源更多。Artix-7 100T 的特征如下:

- 15,850 slices , 每个包含4个6输入LUTs 及8个flip-flops
- 4,860 Kbits 的高速block RAM
- 6个时钟管理模块 (CMT) 每个包含1个混合模式时钟管理器 (MMCM) 及一个锁相环 (PLL)
- 500MHz+ 时钟

外围设备:

- 16个拨码开关
- 16个 LED
- 2个4位7段数码管
- USB-UART接口
- 2个3色LED
- 1个micro SD卡槽
- 12位VGA输出
- 1个单声道PWM音频输出
- 1个PDM麦克风
- 1个3轴加速度计
- 1个温度传感器
- 10/100M以太网 PHY
- 128MB DDR2
- 串行Flash存储器
- 4个Pmod接口
- 1个XADC模拟信号数字化Pmod 接口
- Digilent USB-JTAG接口, 用于FPGA编程, 也可以进行通信
- 1个USB HID Host接口, 可以接鼠标, 键盘和U盘



序号	描述	序号	描述
1	选择供电跳线	13	FPGA 配置复位按键
2	UART/ JTAG 共用 USB 接口	14	CPU 复位按键 (用于软核)
3	外部配置跳线柱(SD / USB)	15	模拟信号 Pmod 端口(XADC)
4	Pmod 端口	16	编程模式跳线柱
5	扩音器	17	音频连接接口
6	电源测试点	18	VGA 连接口
7	16 个 LED	19	FPGA 编程完成 LED
8	16 个拨键开关	20	以太网连接口
9	8 位 7 段数码管	21	USB 连接口
10	可选用于外部接线的 JTAG 端口	22	(工业用) PIC24 编程端口
11	5 个按键开关	23	电源开关
12	板载温度传感器	24	电源接口

第二章 电路实验 - 数字逻辑设计

1.1 实验：基本 I/O 实验

1.1.1 实验介绍

该实验将指导您使用 Vivado 开发环境在 Nexys4DDR 开发板上创建基于 FPGA 的基本 I/O 驱动实验。您将使用 IP 集成的模块设计功能来配置 ZYNQ PS 和添加 IP 创建硬件系统，以及 SDK 创建一个应用程序来验证设计的功能。

1.1.2 实验目标

- 熟悉 Vivado 的使用；
- 熟悉 HDL 编程；
- 掌握如何下载 Bit 文件到 FPGA；
- 熟悉 Nexys4DDR 的硬件解扣子

1.1.3 实验环境

类型	名称	说明
硬件	Nexys4DDR	Artix-7 FPGA 开发板
	USB 下载线	标准 A 型口转 Micro-B 型口
	5V 电源适配器	220V 交流转 5V 直流输出
软件	Vivado 2014.3.1	

表 1-1 实验所需软硬件列表

1.1.4 实验流程

- 创建一个新工程
- 创建一个新的 VHDL 模块
- 输入基本代码
- 设计综合 (Synthesis)
- 创建约束文件

- 生成 FPGA 配置文件
- 下载 FPGA 配置文件
- 在硬件平台上进行功能测试

1.1.5 实验步骤

Step1 : 创建一个新工程

双击桌面的 Vivado 2014.3.1 应用程序图标启动 Vivado 设计环境：



图 1 启动 vivado

点击 Create New Project 建立新工程，点击 Next 继续：

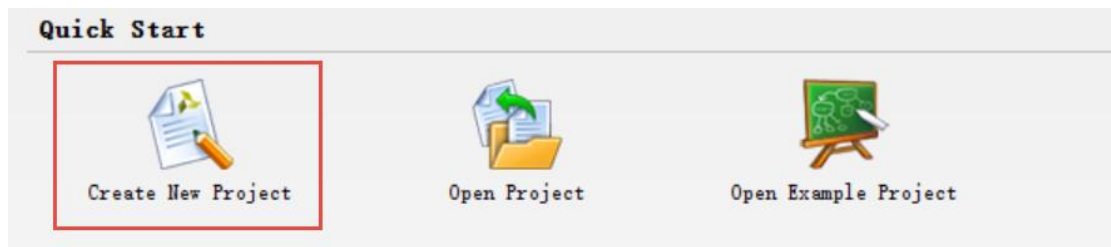


图 2

输入工程名，勾选 Create project subdirectory，点击下一步：

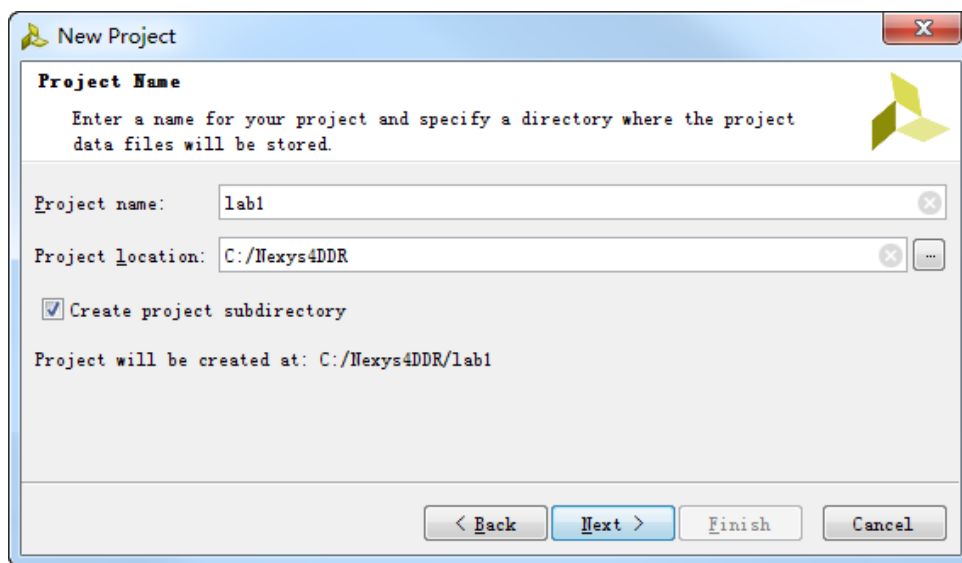


图 3

选择 RTL Project , 勾选 Do not specify source at this time , 点击 Next 继续 :

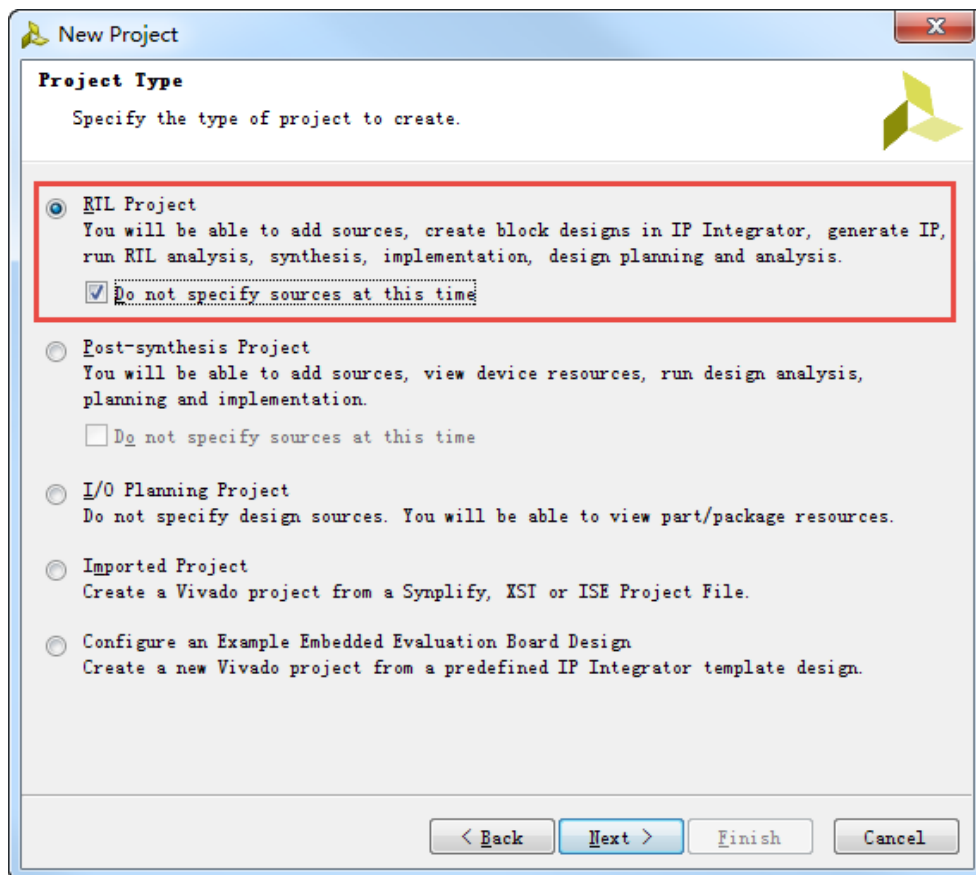


图 4

接着选择 Parts，依次在 Filter 的下拉菜单中选择如下设置：

Product Category	All
Family	Artix-7
Sub-Family	Artix-7
Package	csg324
Speed grade	-1
Temp grade	C

然后在列表中选择 xc7a100tcsg324-1，点击 Next 继续：

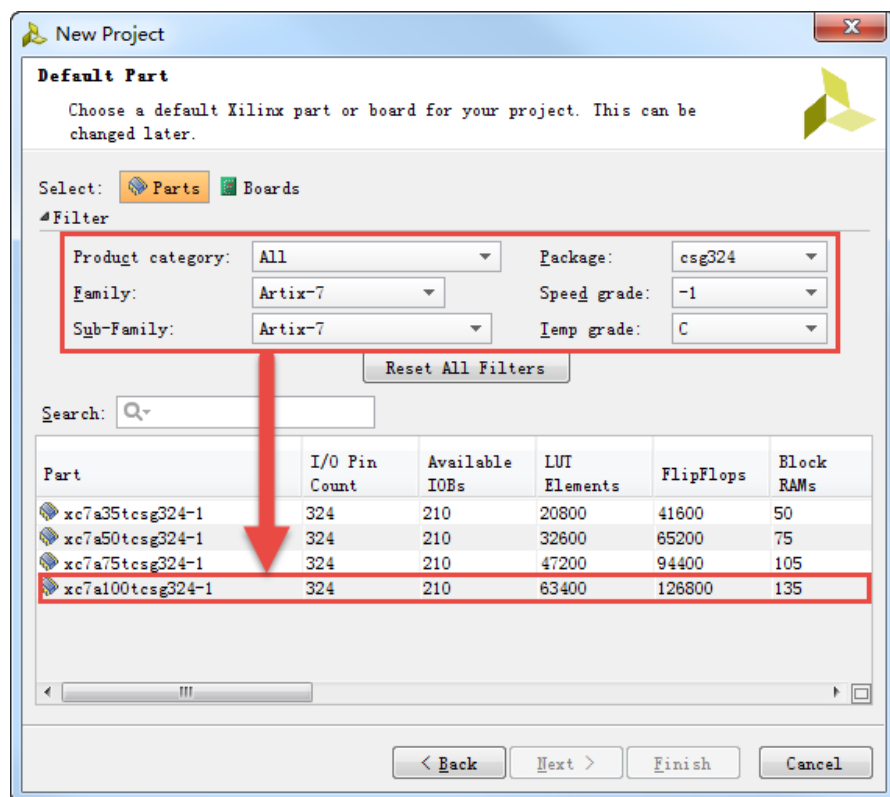


图 5

点击 Finish 完成新工程的创建，如下图所示。



图 6

Step2 : 创建一个新的 VHDL 模块

在 Vivado 主界面的 Project Manager 区域，选中 Design Sources，鼠标右键单击，在弹出的下来菜单中选择 Add Sources：

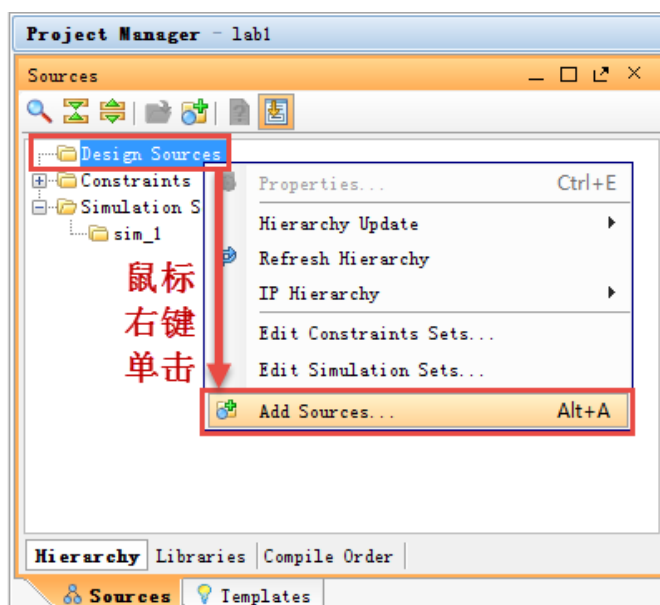


图 7

在添加文件类型中选择 Add or Create Design Sources , 点击 Next :

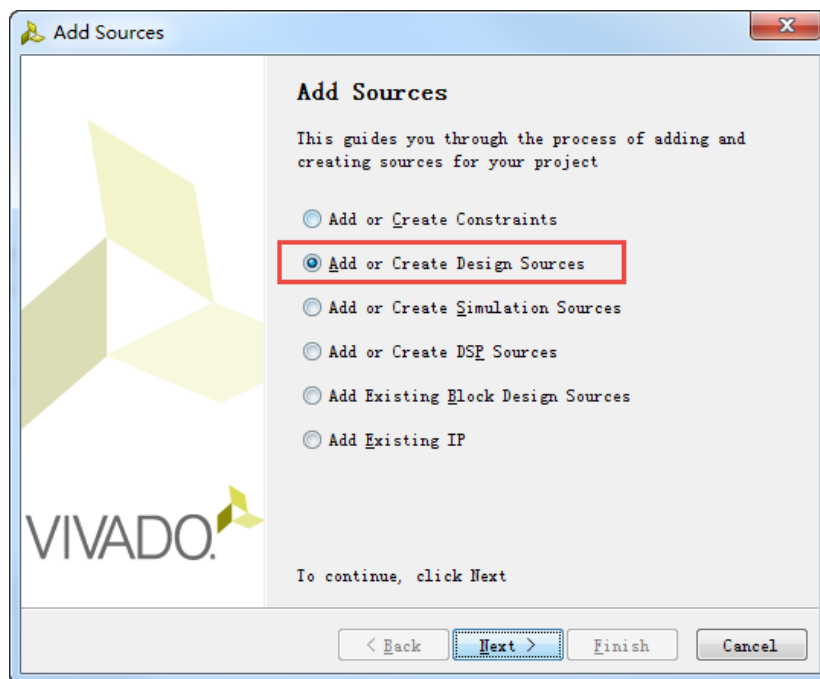


图 8

点击 Create File :

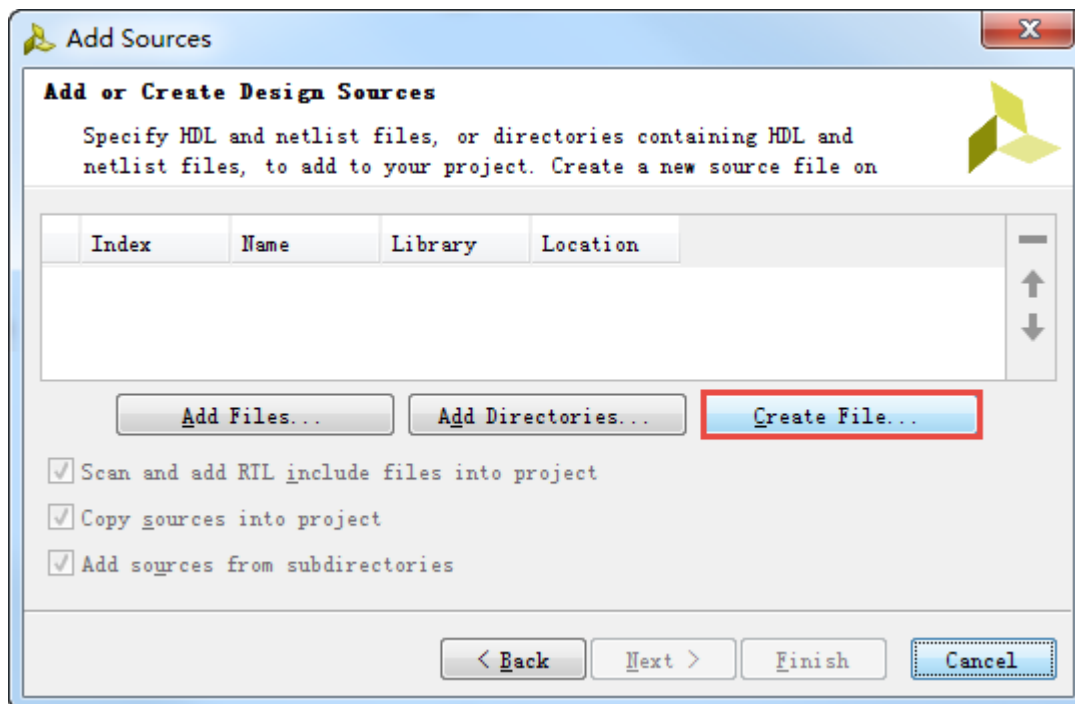


图 9

选择 File type 为 VHDL , File name 一栏输入 : Switch_LEDs , 其他设置不变 , 点击 OK 确定 :

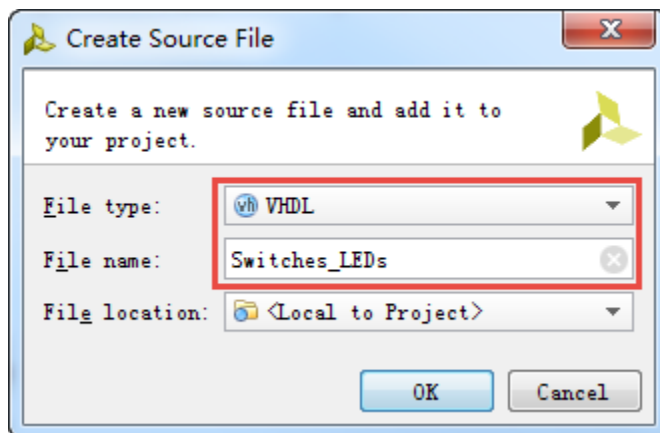


图 10

返回到 Add Sources 窗口栏中 , 保留全部设置 , 直接点击 Finish 完成 Switches_LEDs 文件的添加 :

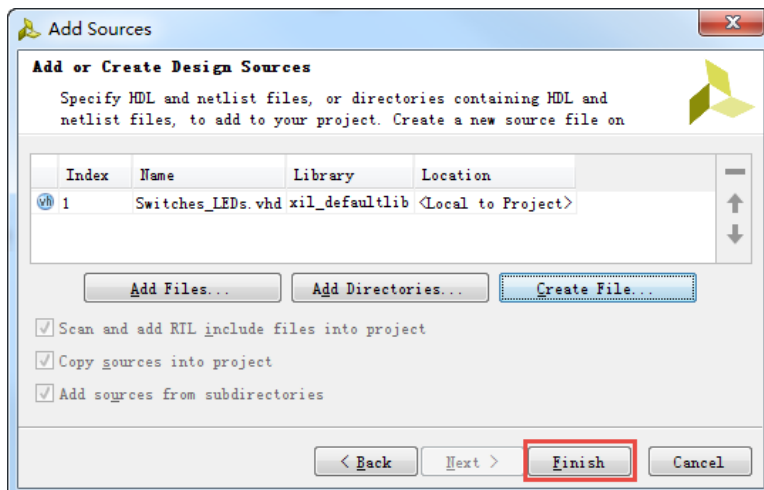


图 11

接着在 Define Module 窗口中 , 输入如下端口设置 :

Port Name	Direction
switch_0	in
switch_1	in
LED_0	out
LED_1	out

点击 OK 完成新 VHDL 模块的创建。

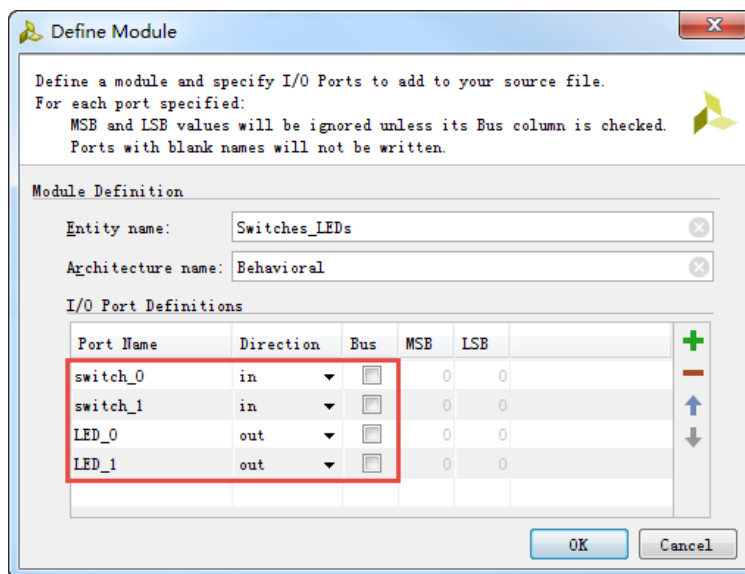


图 5

Step3 : 输入基本代码

在 Project Manager 区域中，双击刚刚添加的 Switches_LEDs.vhd 文件，进入编辑模式：

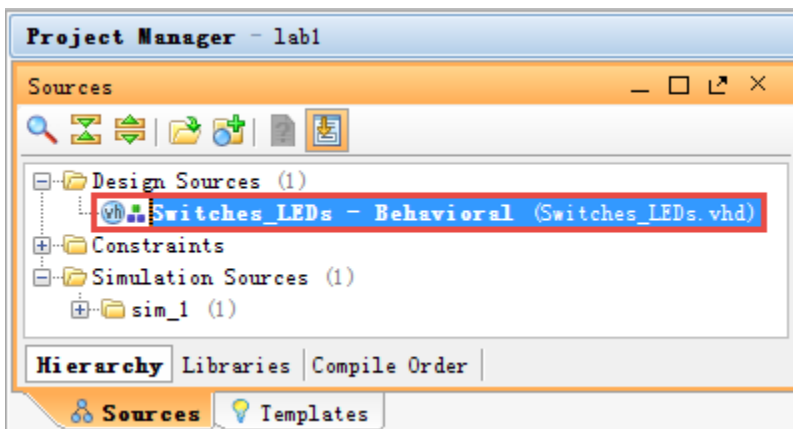


图 6

在文本编辑模式下，在 Switches_LEDs.vhd 文件的大约 44 行开始，输入如下两行以 VHDL 句法描述的电路逻辑行为的语句：

```
LED_0 <= switch_0;
```

```
LED_1 <= switch_1;
```

输入完成后的代码如下图所示，点击保存按钮完成 VHDL 模块的创建。

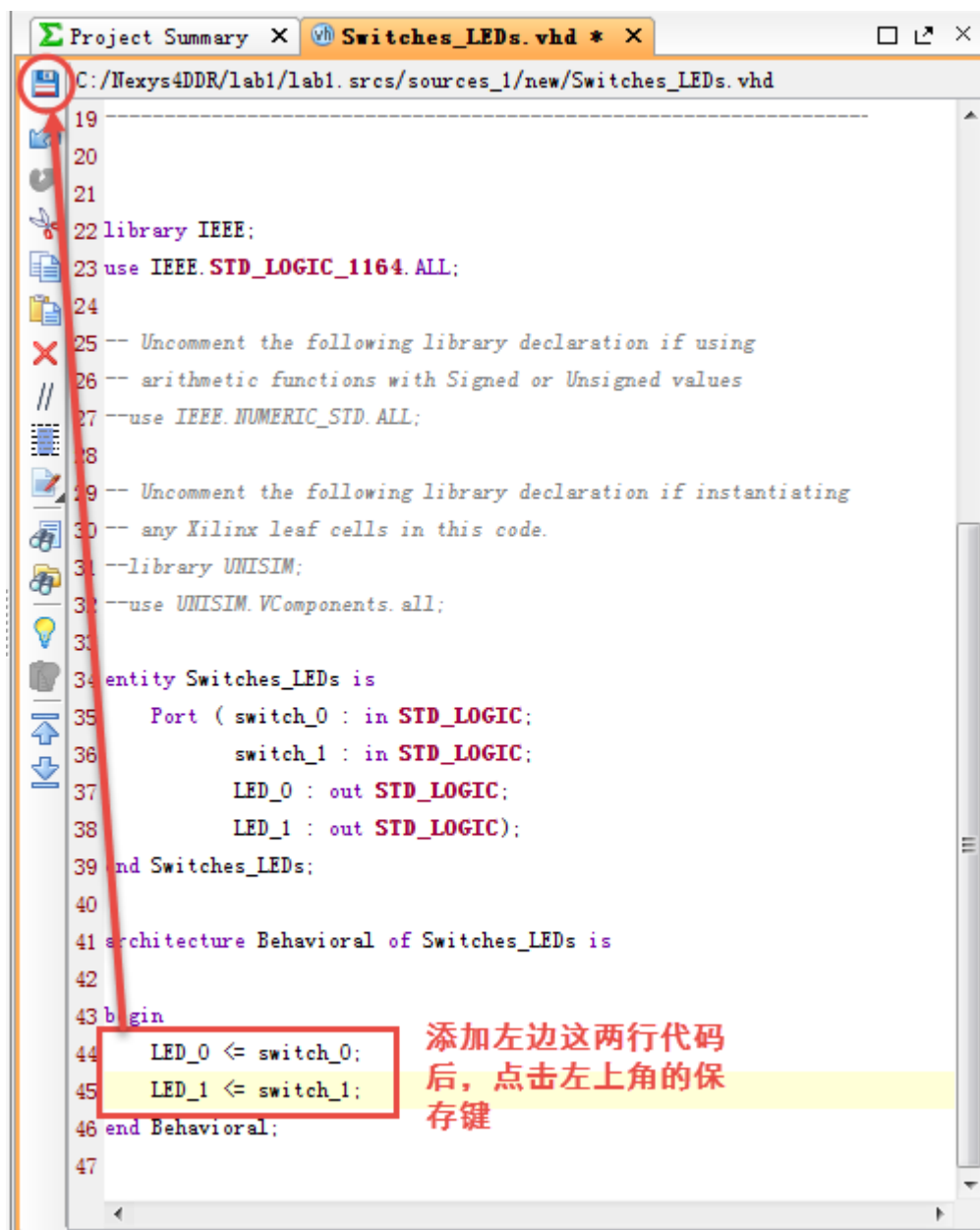


图 14

Step4 : 设计综合 (Synthesis)

在 Vivado 主界面的 Flow Navigator 区域，点击 Run Synthesis，进行 VHDL 模块的综合：

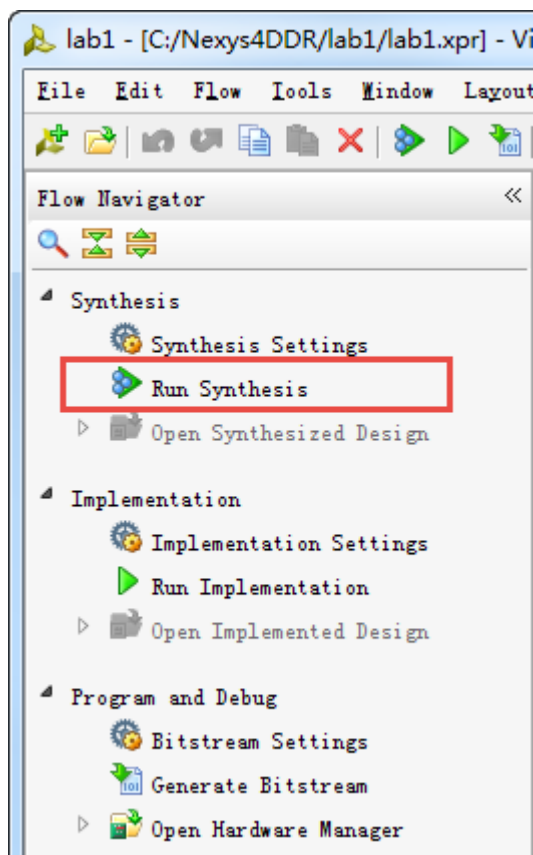
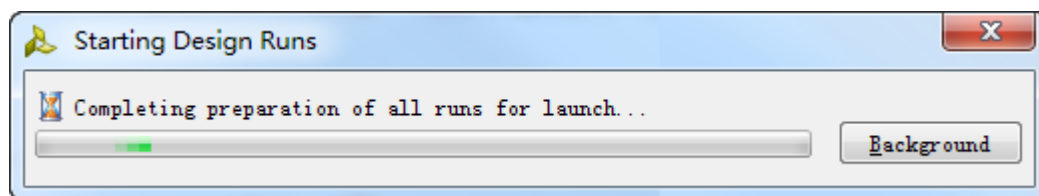


图 15

接着会弹出如下提示窗口，表明此时已进入综合模式：



由于本实验涉及的模块设计非常简单，所以综合过程很快就结束了，综合完成后将弹出如下窗口，点击 Cancel 完成综合。

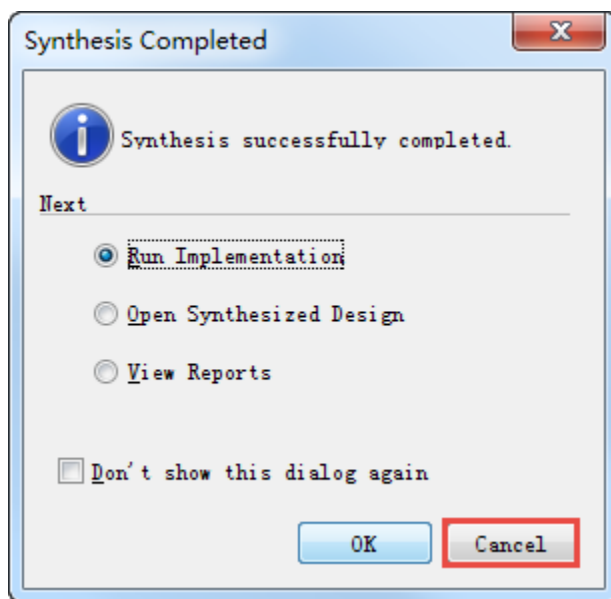


图 16

Step5 : 创建约束文件

现在我们需要添加一个约束文件到工程里面，与 **Step2** 步骤类似，在 Vivado 主界面的 Project Manager 区域，选中 Design Sources，鼠标右键单击，在弹出的下来菜单中选择 Add Sources：

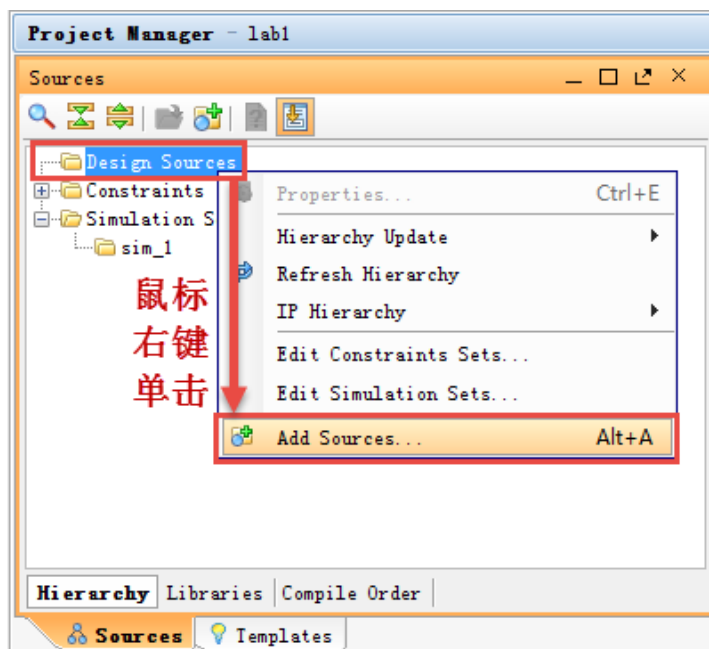


图 7

在添加文件类型中选择 Add or Create Constraints , 点击 Next :

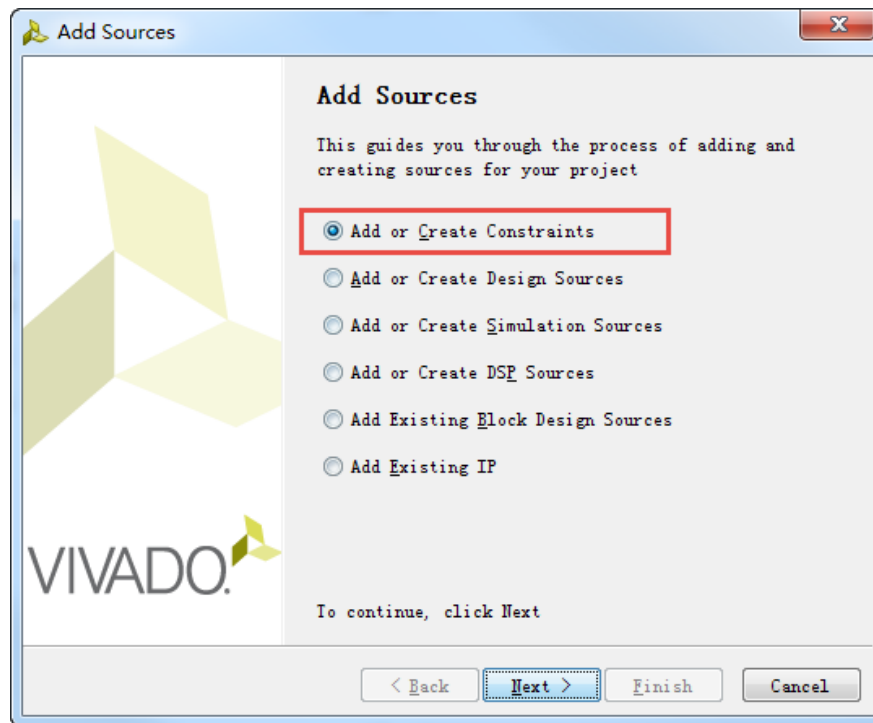


图 8

接着在添加约束文件的窗口点击 Create File :

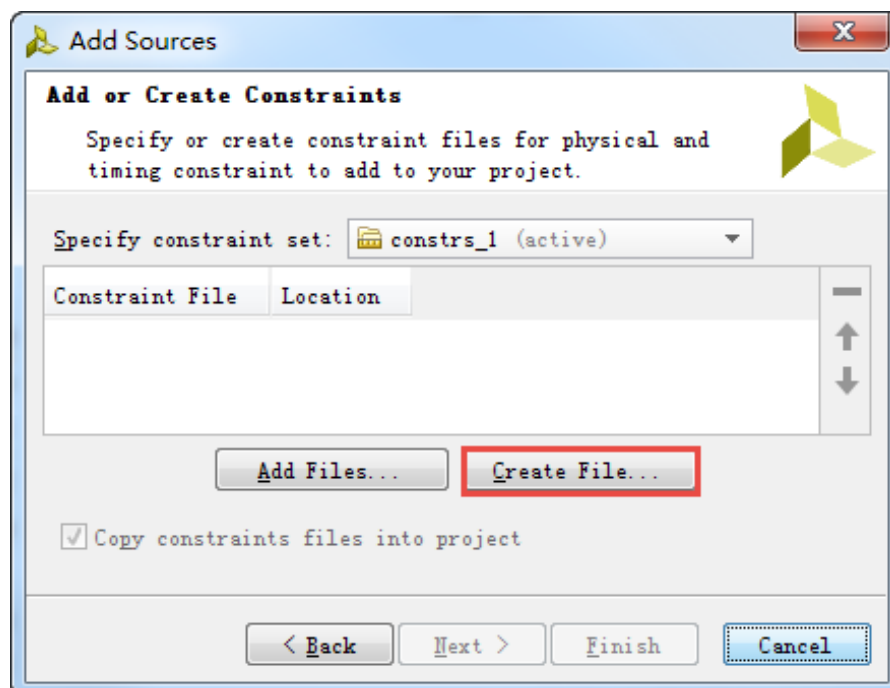


图 9

在新弹出的窗口中，在 File name 一栏输入约束文件名：Switches_LEDs，其他设置不用改变，点击 OK 确定：

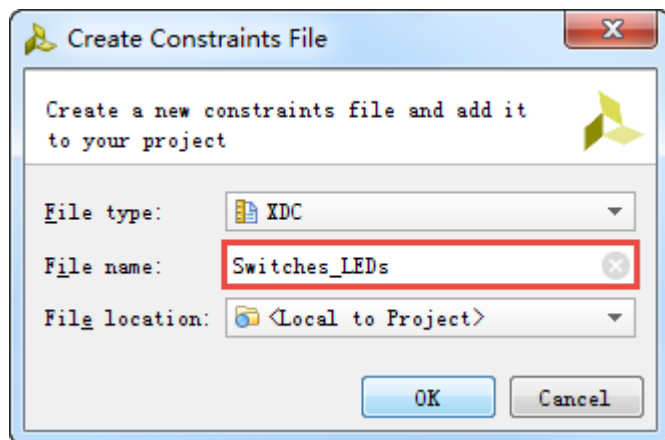


图 10

在返回到添加约束文件窗口时，确认和下图设置一样，确认无误后，点击 Finish。

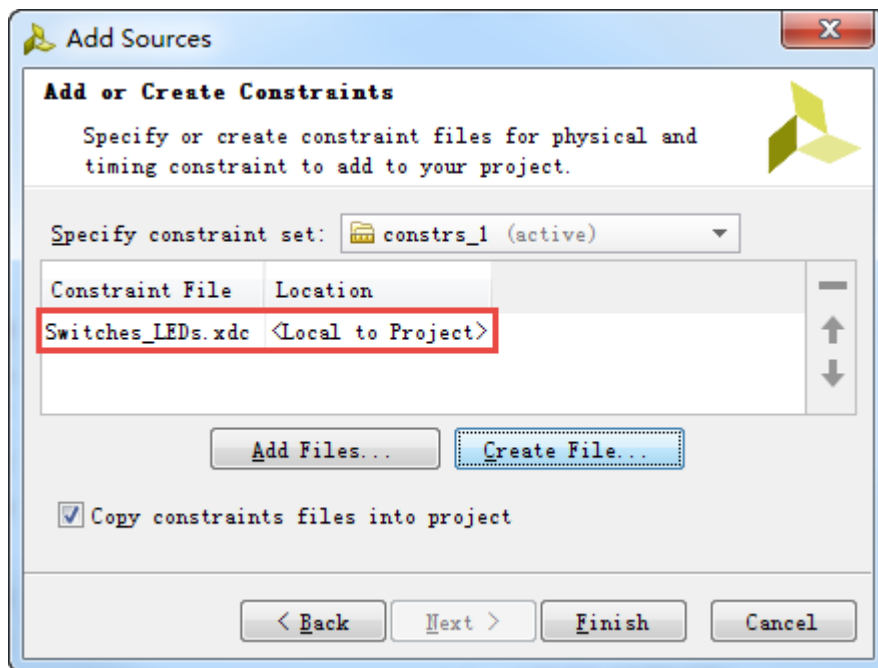


图 11

在 Project Manager 区域，展开 Constrains 选项，直到看见刚才创建的 Switches_LEDs.xdc 文件，双击此文件进入编辑模式：

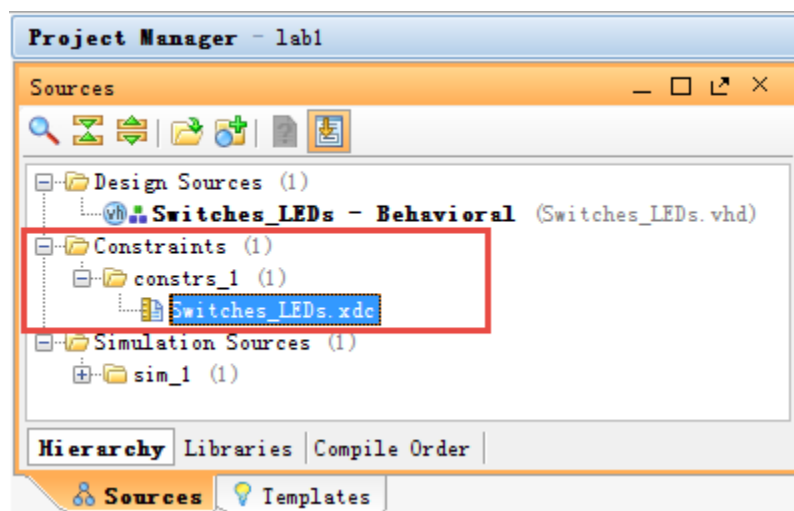


图 12

由于该 XDC(Xilinx Design Constraint)是我们刚刚创建的，此时还是一个空白的文本文件，将如下约束语句添加到该文件中：

```
set_property PACKAGE_PIN J15 [get_ports switch_0]
set_property PACKAGE_PIN L16 [get_ports switch_1]
set_property PACKAGE_PIN H17 [get_ports LED_0]
set_property PACKAGE_PIN K15 [get_ports LED_1]
set_property IOSTANDARD LVCMOS33 [get_ports switch_0]
set_property IOSTANDARD LVCMOS33 [get_ports switch_1]
set_property IOSTANDARD LVCMOS33 [get_ports LED_0]
set_property IOSTANDARD LVCMOS33 [get_ports LED_1]
```

添加后的效果如下图所示，点击左上角的保存按钮。

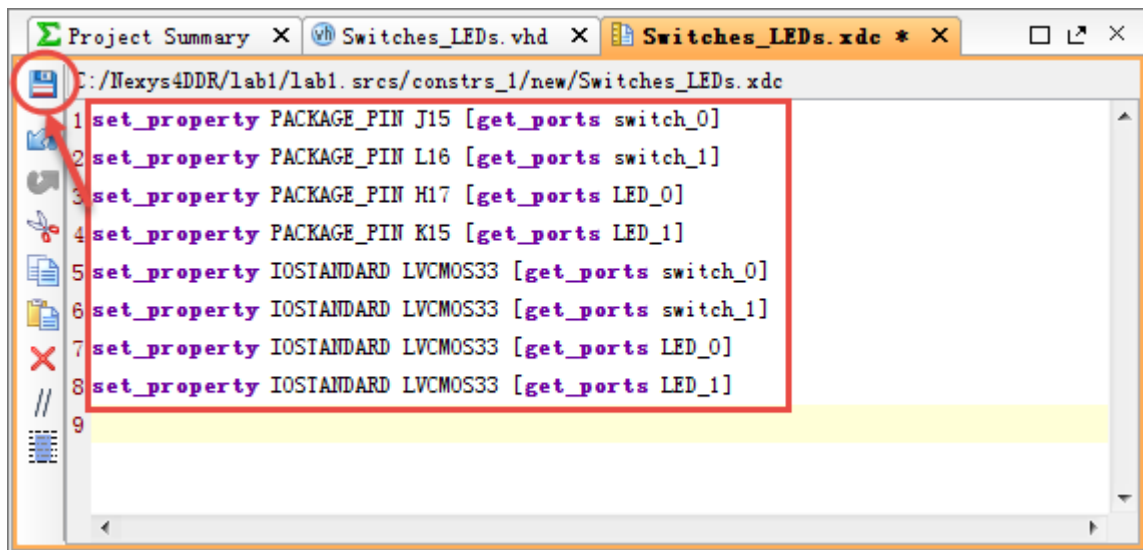


图 13

Step6 : 生成 FPGA 配置文件

接着点击 Flow Navigator 区域 Program and Debug 一栏下的 Generate Bitstream :

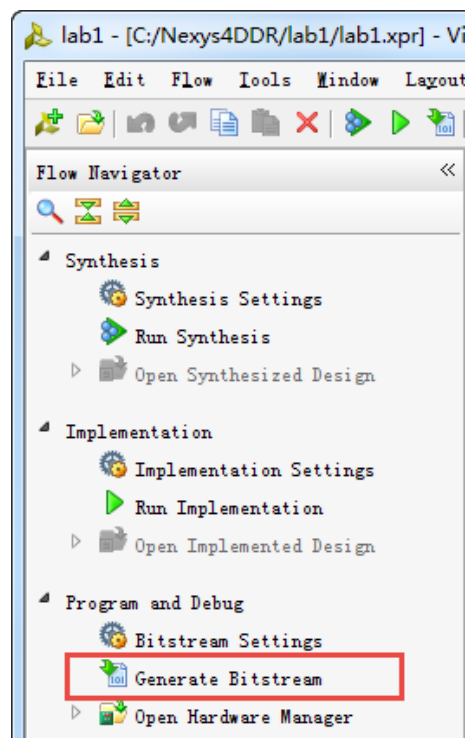


图 14

接着会弹出综合结果过期的提示，询问我们是否重新综合并生成 Bitstream 文件，点击 Yes 继续：

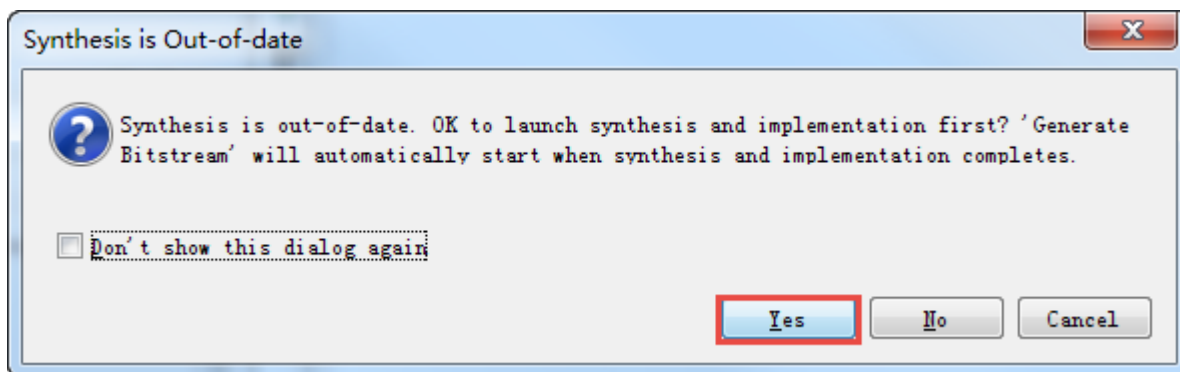


图 15

整个过程结束后，弹出 Bitstream 文件生成的提示窗口，在窗口中选择 Open Hardware Manager 选项，然后点击 OK 进入下载 FPGA 配置文件环节：

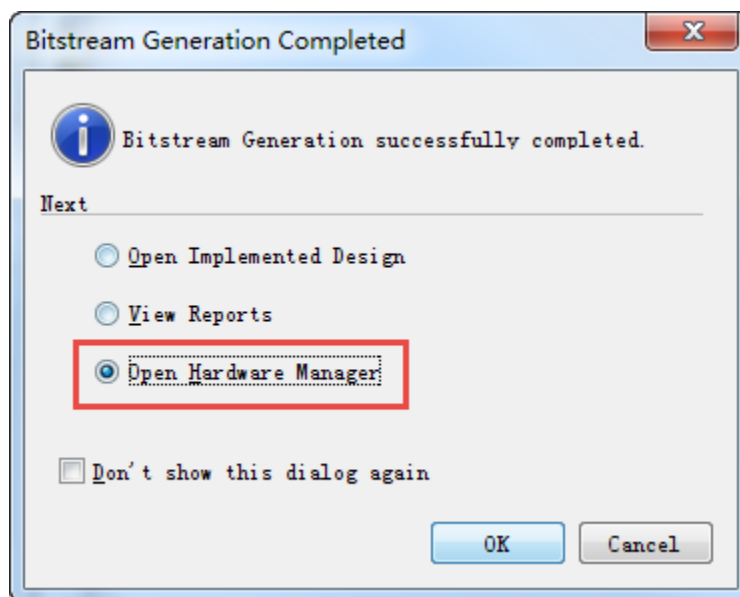


图 16

Step7 : 下载 FPGA 配置文件

在打开 Hardware Manager 之后，进入如下界面：

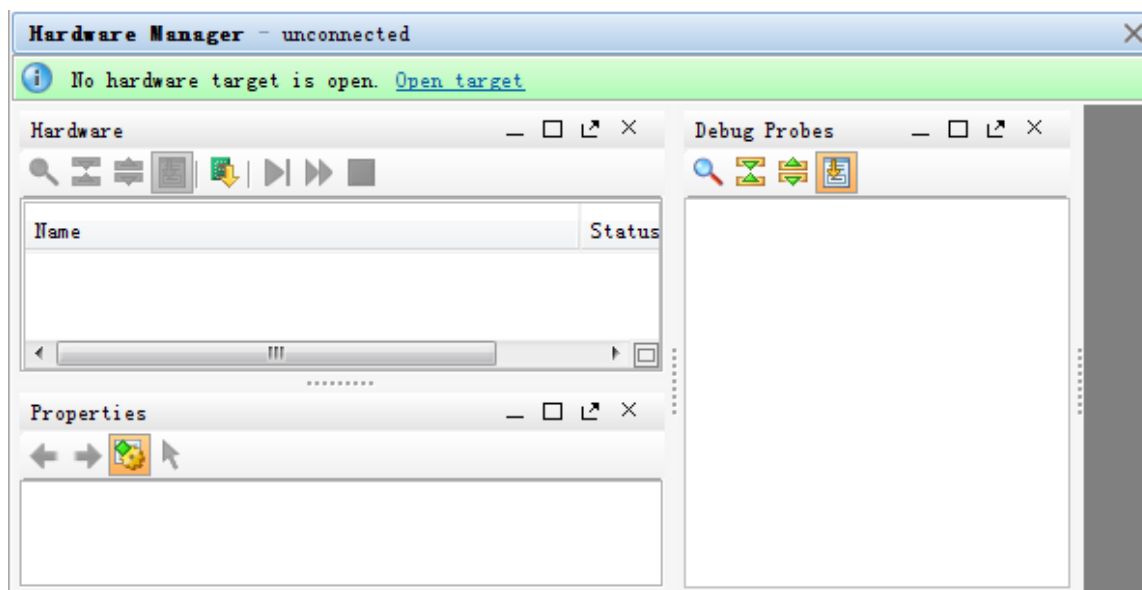


图 17

将 Nexys4-DDR 用 USB 线和电脑连接起来，上电前请确认已经通过 JP3 跳线正确的配置了电源输入方式为 USB 或外接电源（WALL），以及通过 JP1 跳线正确的配置了 JTAG 下载方式，然后打开开关，在全部驱动程序安装完成后（与下图类似），点击 Open target，选择 Auto Connect：

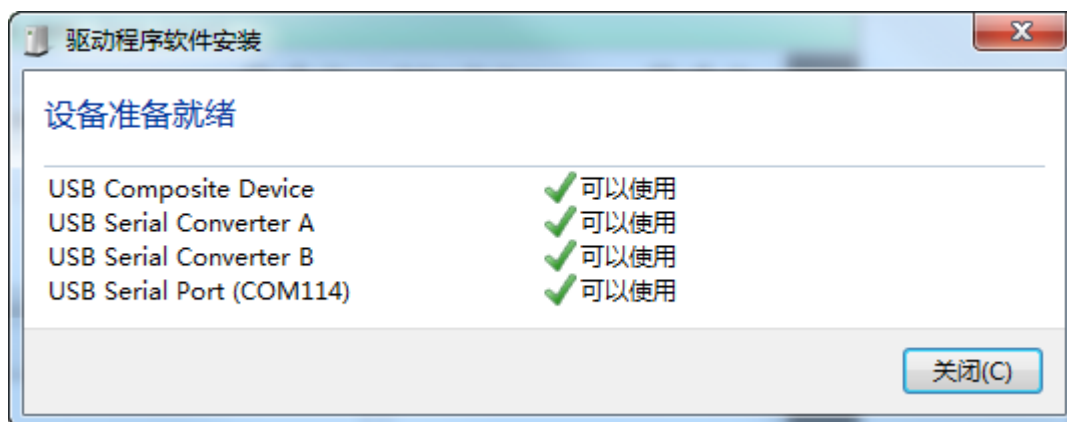


图 18

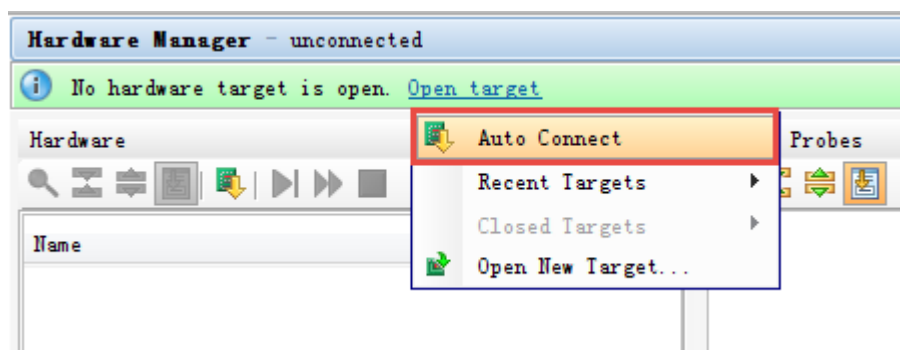


图 19

接着能在 Hardware 窗口看到 Vivado 已经能检测到 Nexys4-DDR JTAG 扫描链上的 Artix-7 100T FPGA。点击 Program device，在弹出菜单中单击 xc7a100t_0：

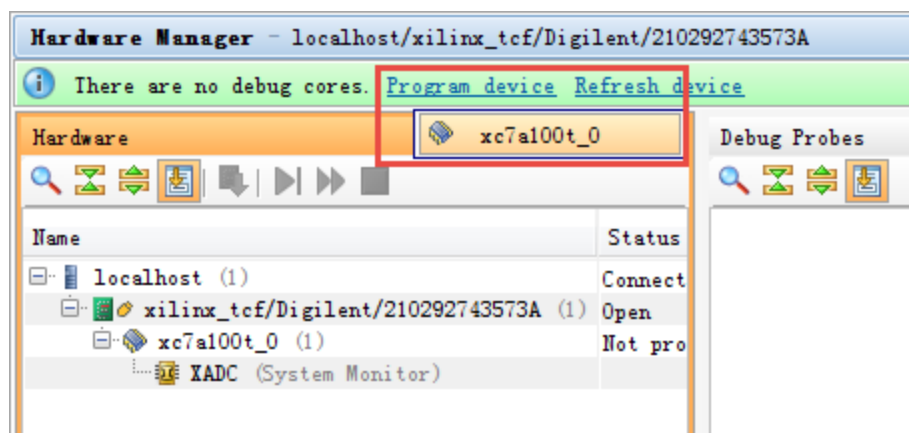


图 20

在弹出窗口中，Vivado 会自动根据当前工程状态，将刚刚生成好的 Switches_LEDs.bit 文件填入选项栏，此时只需点击 Program 按钮：

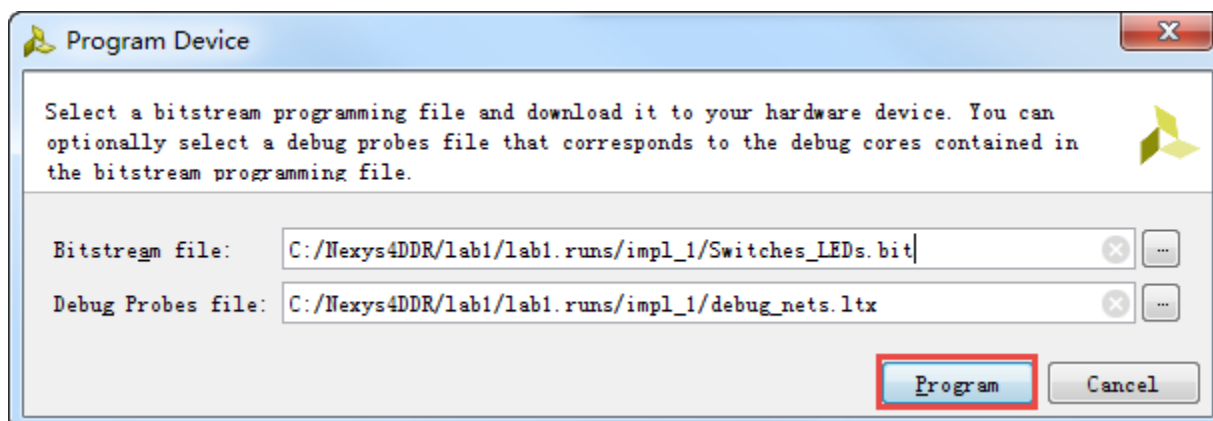


图 21

随后工具开始对 Nexys4-DDR 进行 Program 操作。

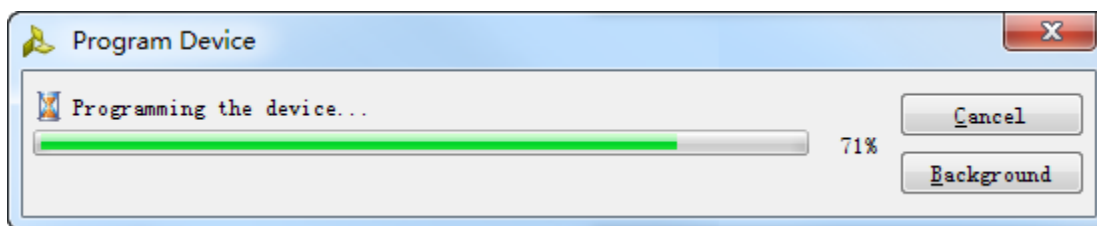


图 22

Step8 : 下载 FPGA 配置文件

等待 Hardware Manager 将 bit 文件烧写进 Nexys4 后，将首先会看到 Nexys4-DDR 上表示烧写完成的 Done 指示会点亮 (LD21)，接着拨动 SW0 和 SW1，观察 LD0 和 LD1 的反应。

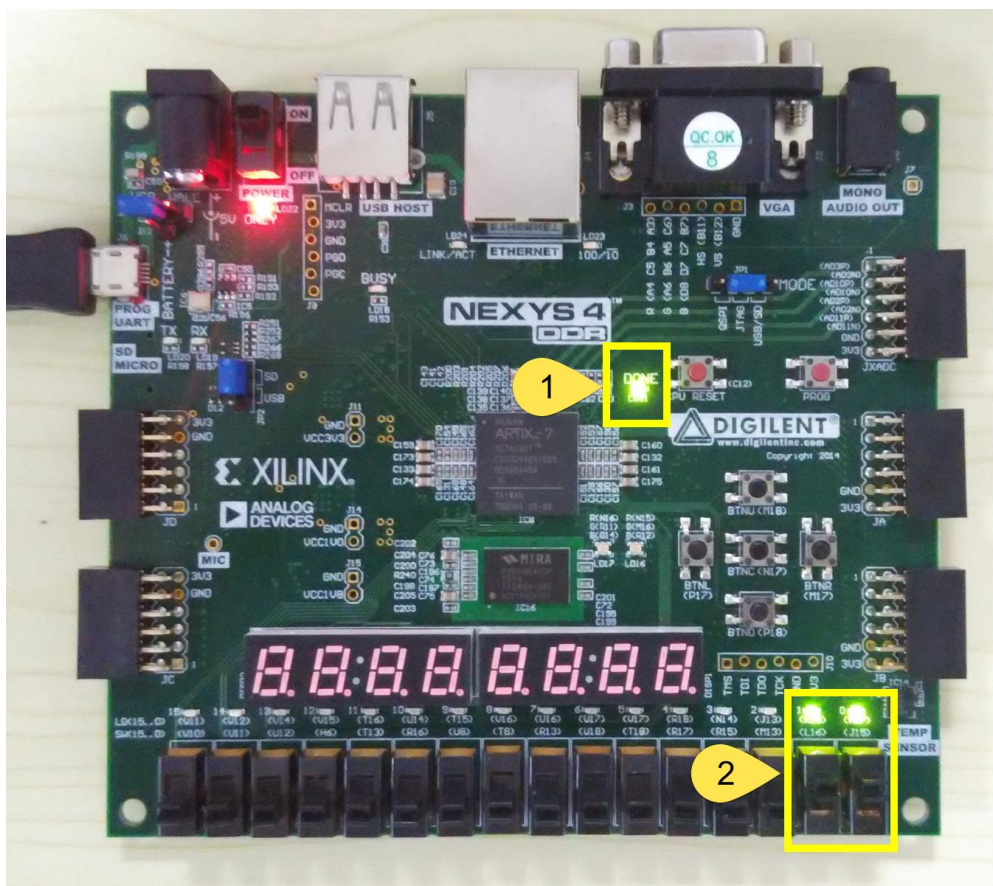


图 23