
1 Vivado 使用说明

在学习并尝试本章节前，你需要具有以下环境和能力：

- (1) 装有 Vivado 的电脑一台。
如果电脑尚未安装 Vivado，请参考文档“A03_Vivado 安装说明”。
- (2) 如果需要上板运行，则需要龙芯体系结构教学实验箱（Artix-7）一套。
实验箱介绍，请参考文档“A01_龙芯体系结构教学实验箱（Artix-7）介绍”。
- (3) 如果需要编写 Verilog，则需要一定的规范的 Verilog 编程能力。
请自学 Verilog。

通过本章节的学习，你将获得：

- (1) 初步熟悉龙芯 Artix-7 实验箱。
- (2) 基本的 Vivado 使用方法。
- (3) 初步了解 Vivado 的 XDC 约束文件。

Vivado 有两种开发模式，Project Mode 和 Non-Project Mode。两者的主要区别是，Project Mode 使用 Flow Navigation 更加自动化，Non-Project Mode 使用 Tcl 命令直接控制操作，更手动但更不受限制。实际中根据开发需求选择这两种模式，一般简单设计中采用 Project Mode。

下面以 led 实验为例介绍如何使用 Vivado 新建一个工程。led 实验实现开发板上 8 个拨码开关控制 8 个 led 灯亮灭。

1.1 数字电路设计的一般流程

在介绍 Vivado 工具的使用前，先简单介绍下数字电路设计的一般流程，如下图（图片来自网络）。图中的最后步骤“Sign off”为物理签核，就是得到最终的电路版图。将电路版图交付晶元代工厂，就可以生产芯片了。

在图 1-1 中用红色标出了 FPGA 开发的流程，FPGA 开发的流程和一般数字电路设计的流程主体部分是对应的。但在最后步骤时，FPGA 开发不是生成电路版图，而是生成 bit 流文件，并下载到 FPGA 开发板上进行运行。

本章的 1.4 节涉及图 1-1 中的第 2、3 步，1.6 节涉及图 1-1 中的第 4、5、6、7、8 步。

1.5 节“添加约束文件”则是综合与布局布线之间的步骤。在图 1-1 中的“Static Timing Analysis”（静态时序分析，简称为 STA）也需要添加约束文件，只是一般数字电路设计中的约束文件只是时序、cell 单元、布局上的约束。而 FPGA 的设计约束除了这些外，还起到引脚绑定的作用。在本学期的实验中，约束文件更多的作用就只是引脚绑定，具体请详细阅读本章 1.5 节。

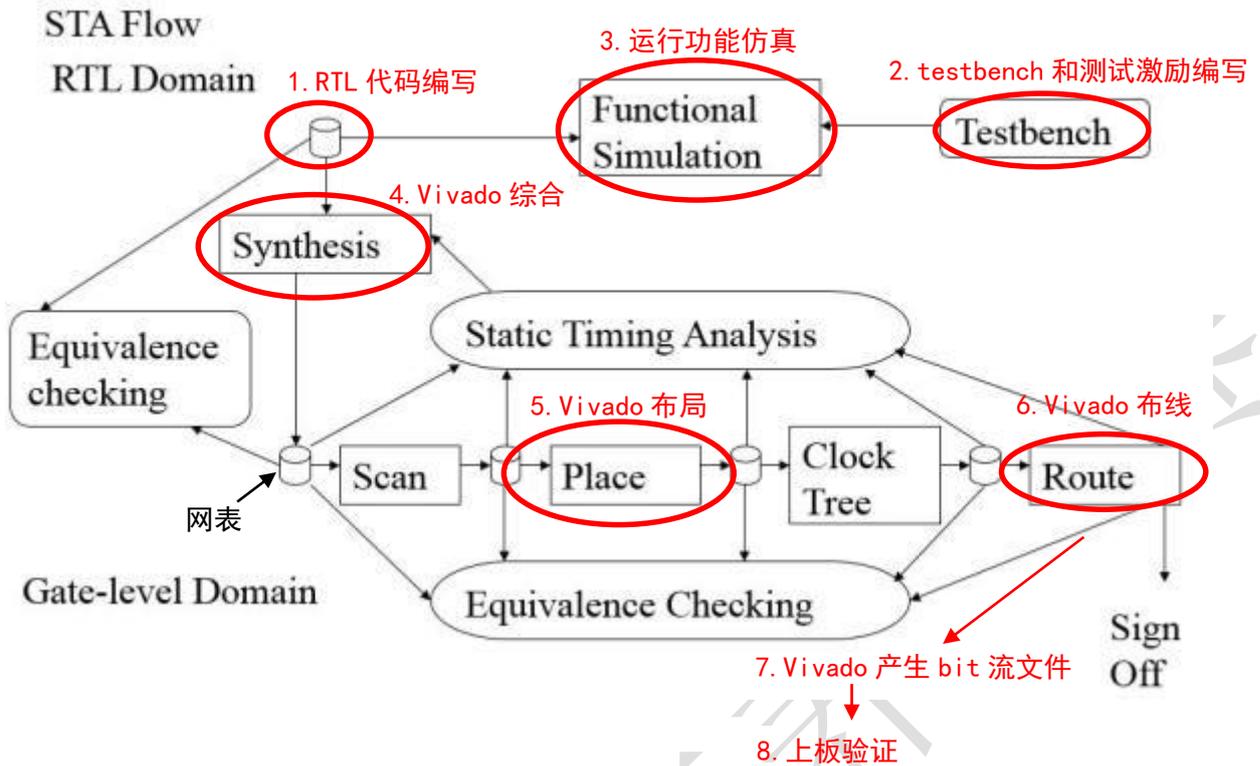


图 1-1 数字电路设计与 FPGA 开发对照图

1.2 Vivado 新建工程

通过双击桌面快捷方式或开始菜单的“Xilinx Design Tools→Vivado 2017.1”打开 Vivado 2017.1，在界面上“Quick Start”下选择“Create Project”。

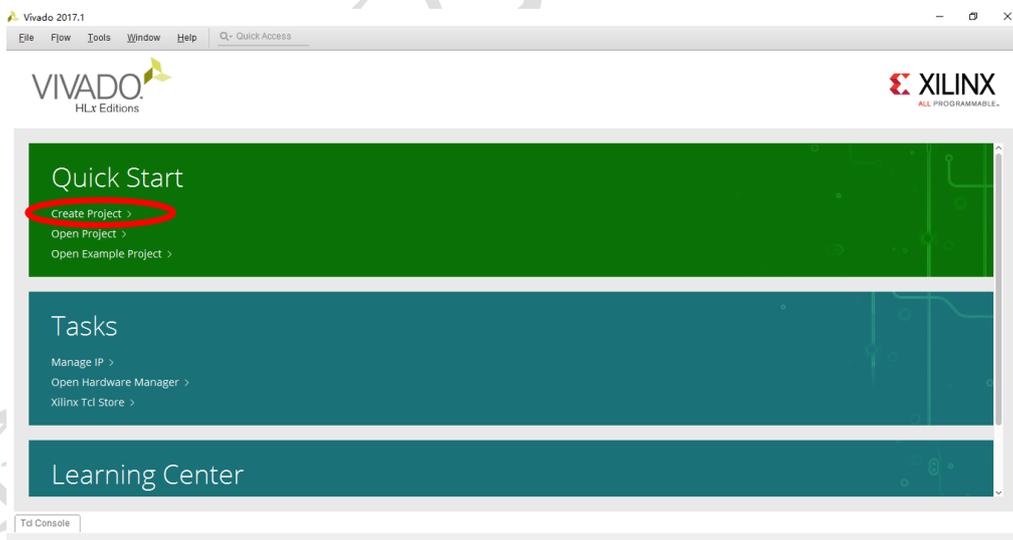


图 1-2 Vivado 启动界面

新建工程向导，点击“Next”。

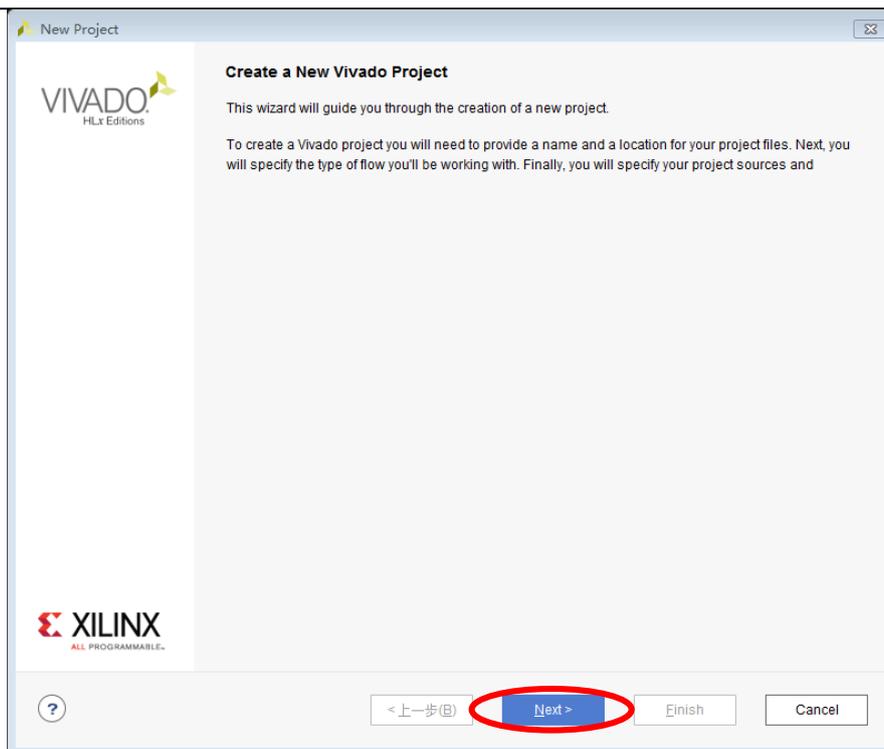


图 1-3 新建工程向导

输入工程名称并选择工程的文件位置，并勾选“Create project subdirectory”选项，为工程在指定存储路径下建立独立的文件夹。设置完成后，点击“Next”。

注意：工程名称和存储路径中不能出现中文和空格，建议工程名称以字母、数字、下划线来组成。

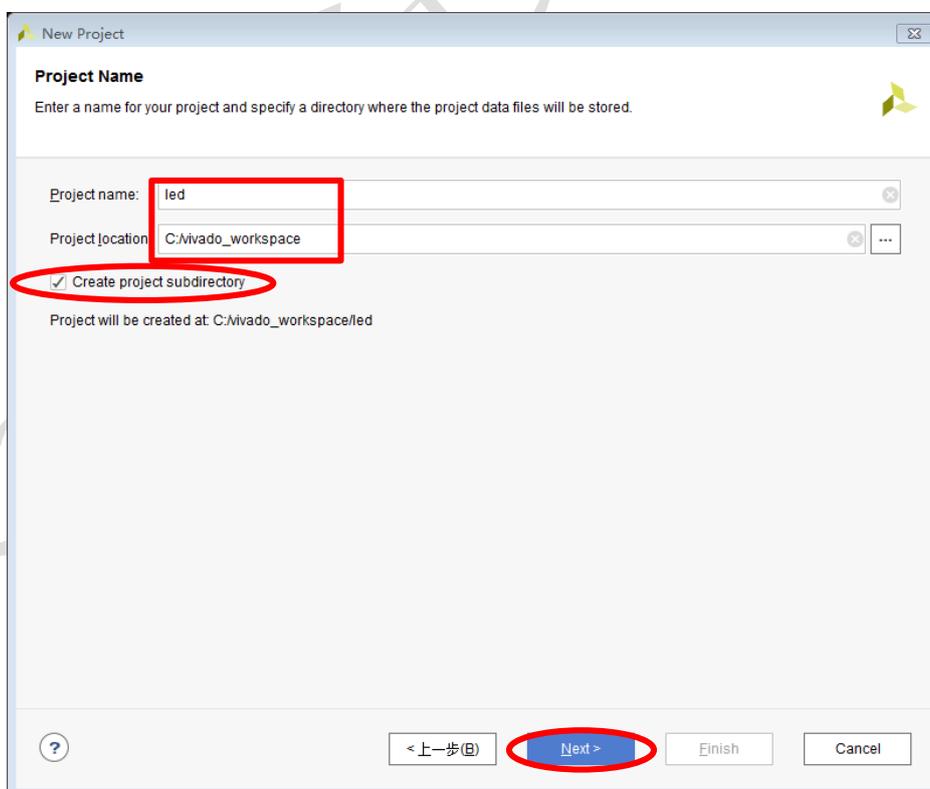


图 1-4 设置工程名称和位置

选择“RTL Project”一项，并勾选“Do not specify sources at this time”，勾选该选项是为了跳过在新建工程的过程中添加设计源文件，如果要在新建工程时添加源文件则不勾选。点击“Next”。

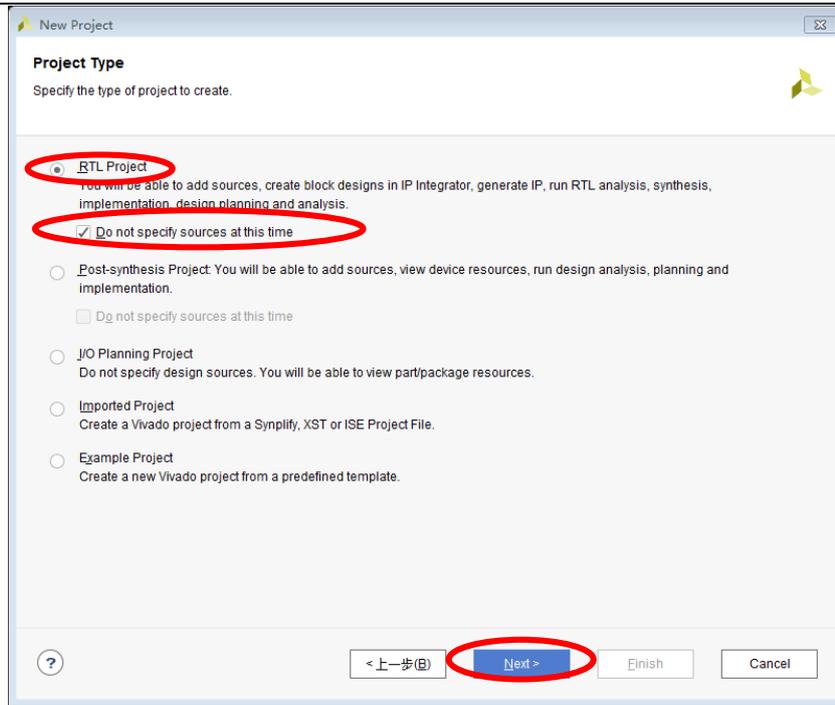


图 1-5 设置工程类型

根据使用的 FPGA 开发平台，选择对应的 FPGA 目标器件。根据实验平台搭载的 FPGA，在筛选器的“Family”选择“Artix 7”，“Package”选择“fbg676”，在筛选得到的型号里面选择“xc7a200tfbg676-2”。点击“Next”。

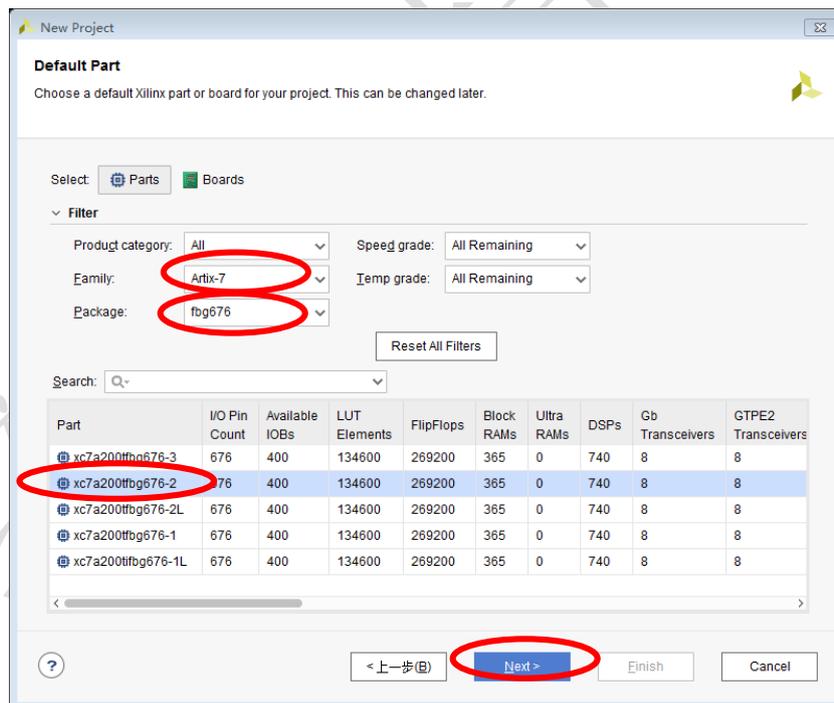


图 1-6 选择目标器件

确认工程设置信息是否正确。正确点击“Finish”，不正确则点击“上一步”返回相应步骤修改。

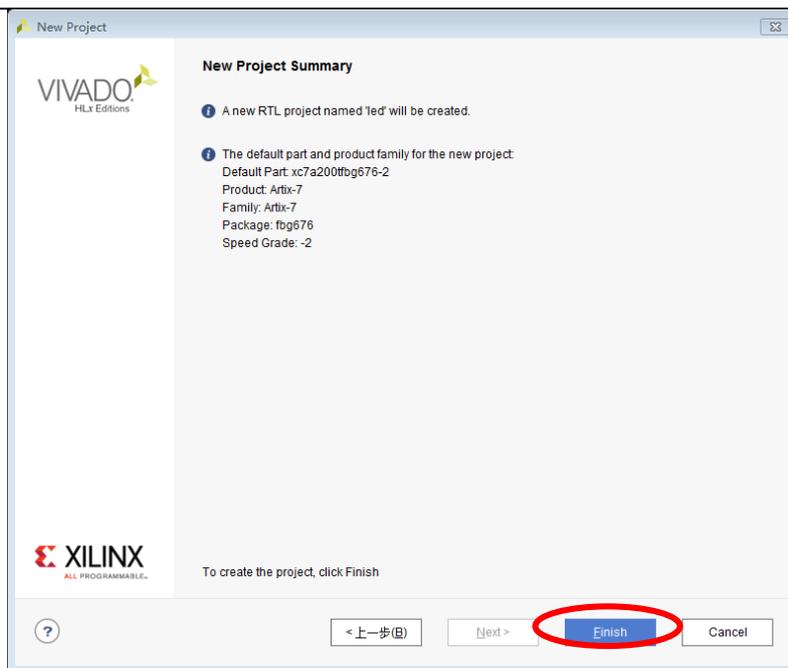


图 1-7 工程信息

完成工程新建。

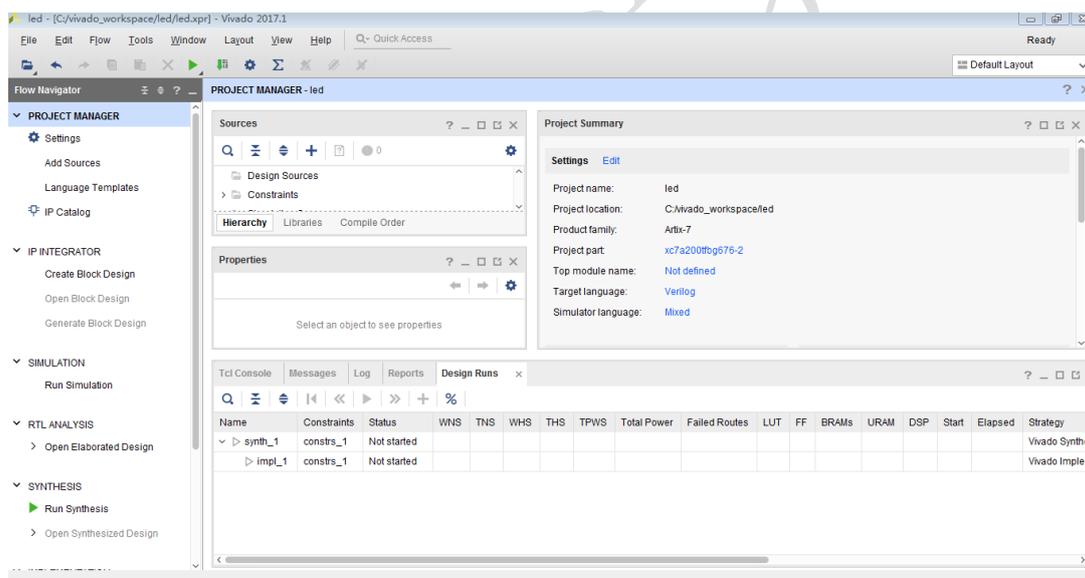


图 1-8 工程新建完成显示界面

1.3 RTL 设计输入

以使用 Verilog 完成 RTL 设计为例。Verilog 代码都是以“.v”为后缀名的文件，可以在其他文件编辑器里写好，再添加到新建的工程中，也可以在工程中新建一个再编辑。

添加源文件。在“Flow Navigator”窗口下的“Project Manager”下点击“Add sources”，或者点击“Source”窗口下“Add Sources”按钮，或者使用快捷键“Alt + A”。

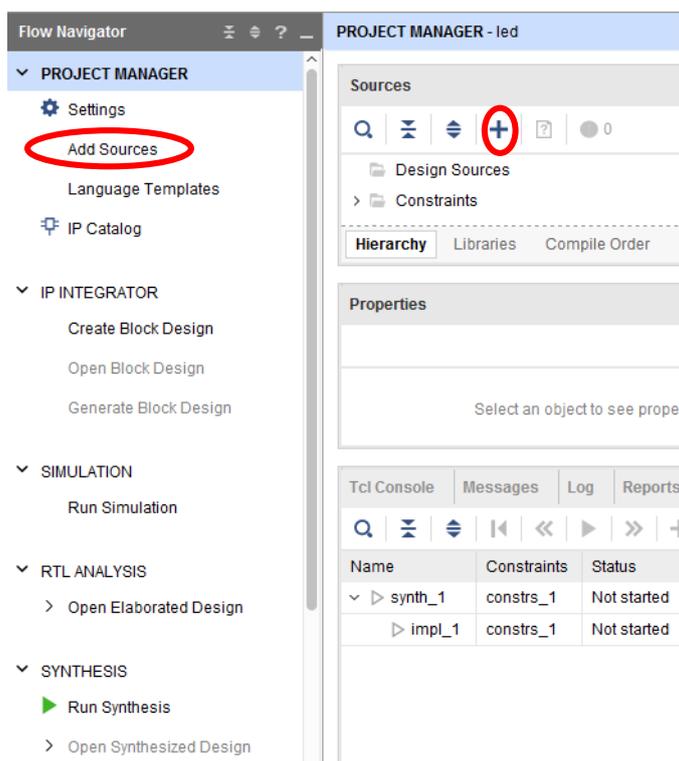


图 1-9 添加源文件

添加设计文件。选择“Add or create design sources”来添加或新建 Verilog 或 VHDL 源文件，点击“Next”。

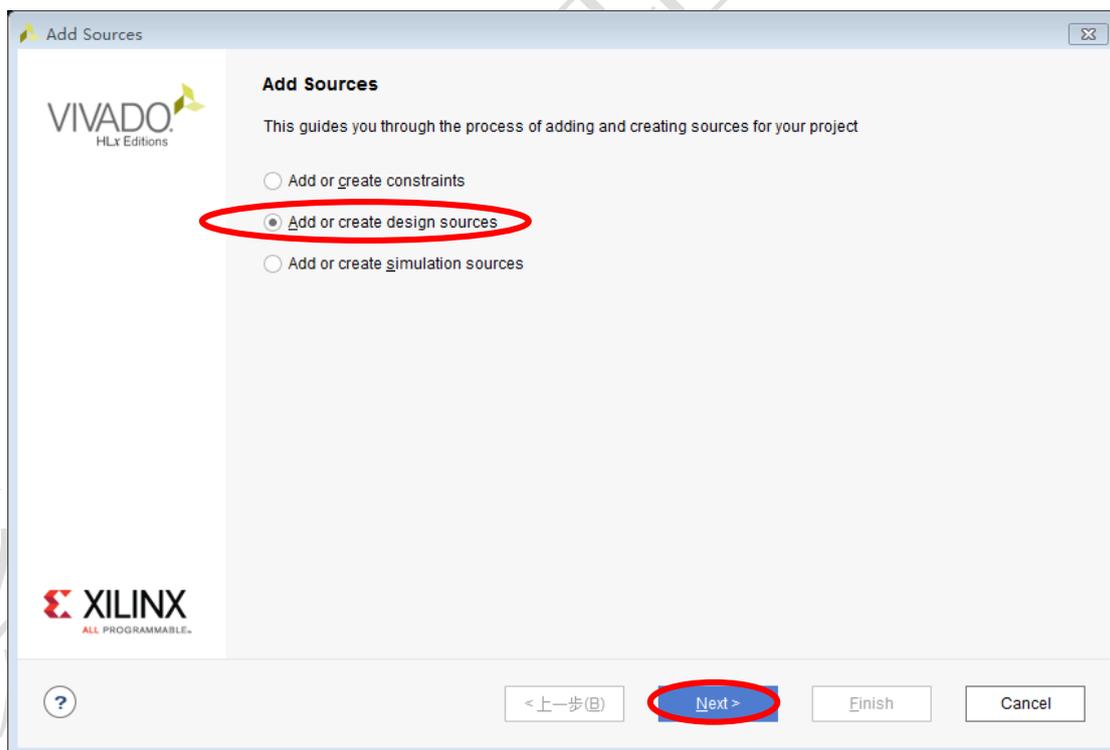


图 1-10 添加设计文件

添加或者新建设计文件。如添加已有设计文件或者添加包含已有设计文件的文件夹，选择“Add Files”或者“Add Directories”，然后在文件浏览窗口选择已有的设计文件完成添加。如创建新的设计文件，则选择“Create File”。这里新建文件。

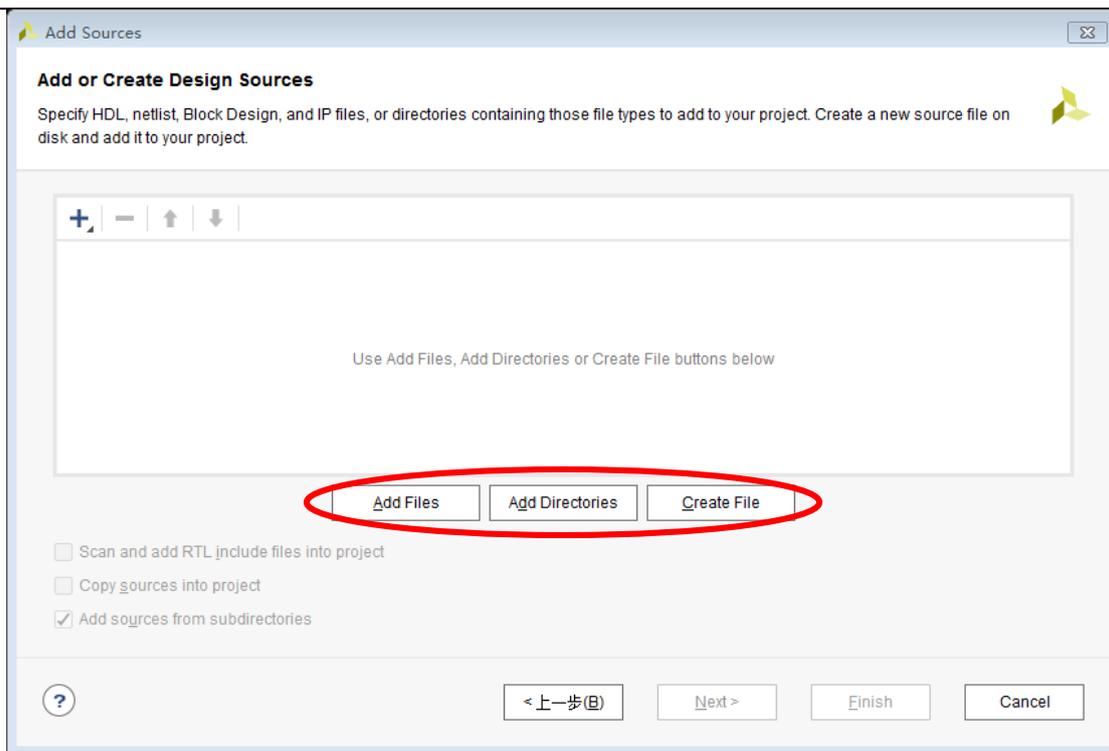


图 1-11 添加或者新建设计文件

设置新创建文件的类型、名称和文件位置。注意：文件名称和位置路径中不能出现中文和空格。

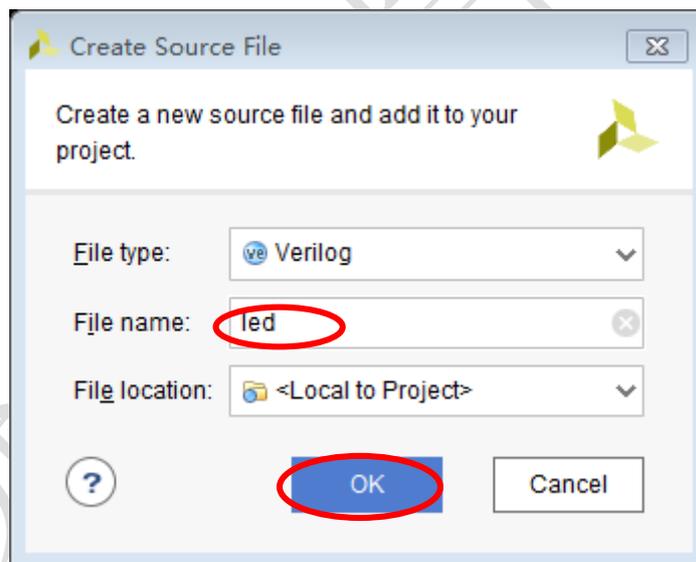


图 1-12 设置新的设计文件

继续添加设计文件或者修改已添加设计文件设置。点击“Finish”。

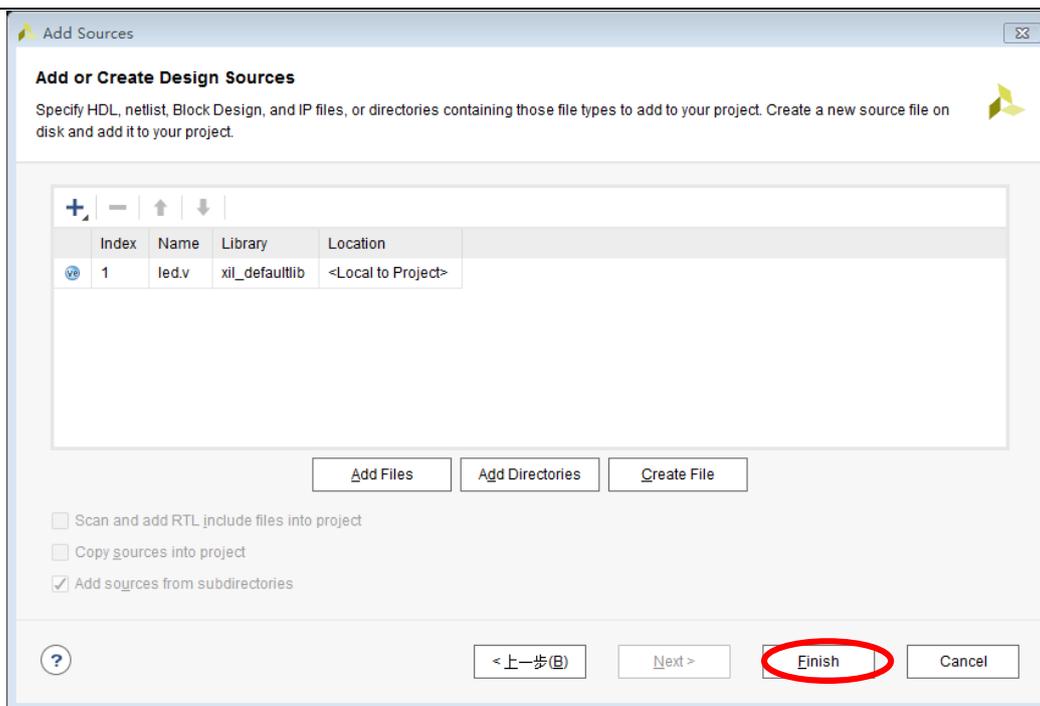


图 1-13 完成设计文件添加

模块端口设置。在“Module Definition”中的“I/O Port Definitions”，输入设计模块所需的端口，并设置端口方向，如果端口为总线型，勾选“Bus”选项，并通过“MSB”和“LSB”确定总线宽度。完成后点击“OK”。端口设置也可以在编辑源文件时完成，即可以在这一步直接点“OK”跳过。

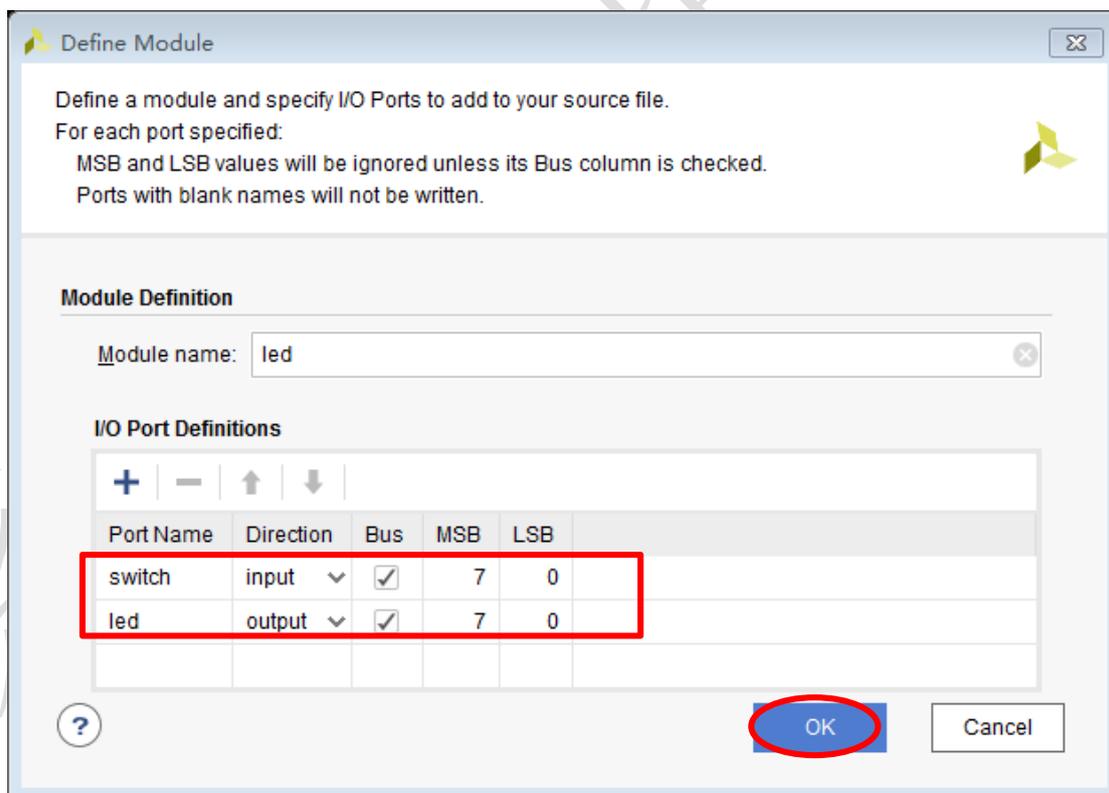


图 1-14 模块端口设置

双击“Sources”中的“Design Sources”下的“led.v”中打开该文件，输入相应的设计代码。如果设置时文件位置按默认的<Local to Project>，则设计文件位于工程目录下的“\led.srcs\sources_1\new”中。完成的设计文件如

下图所示。

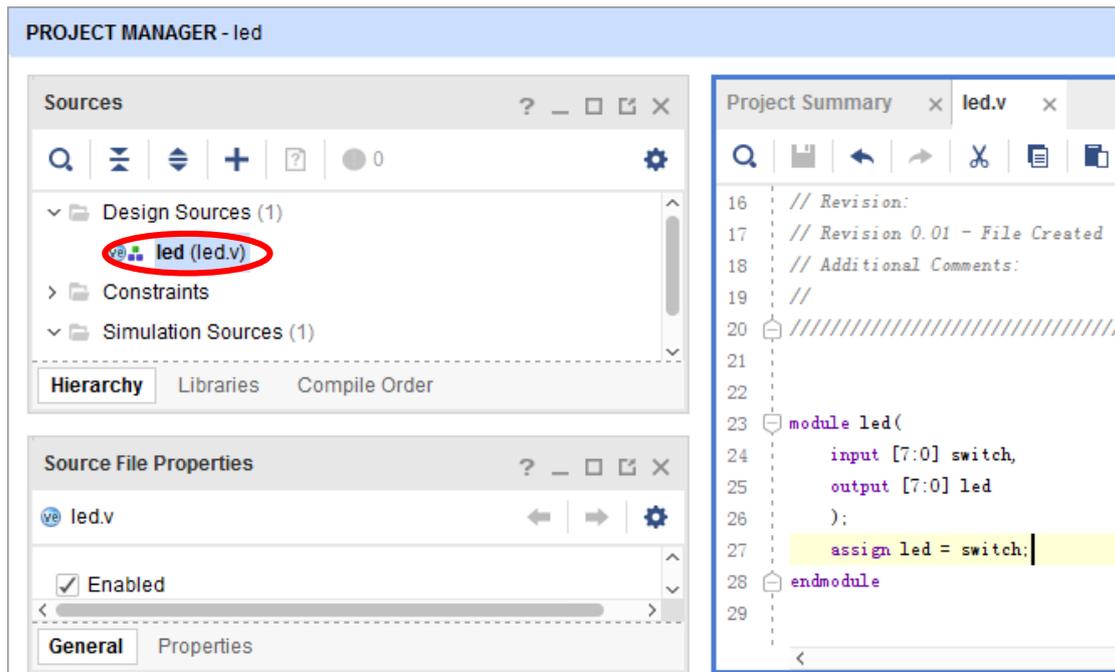


图 1-15 编辑设计文件

1.4 功能仿真

Vivado 集成了仿真器 Vivado Simulator，这里介绍使用 Vivado 仿真器仿真的方法。也可以使用 ModelSim 进行仿真。

首先添加测试激励文件。在“Source”中“Simulation Sources”右击选择“Add source”。

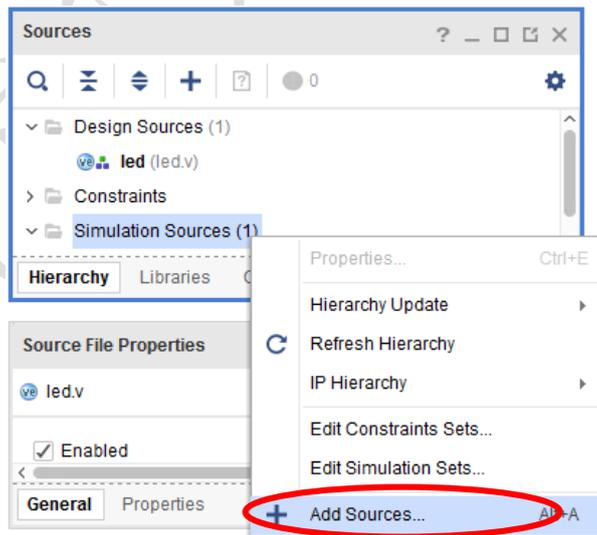


图 1-16 添加测试激励文件

在“Add Source”界面中选择“Add or Create Simulation Sources”，点击“Next”。

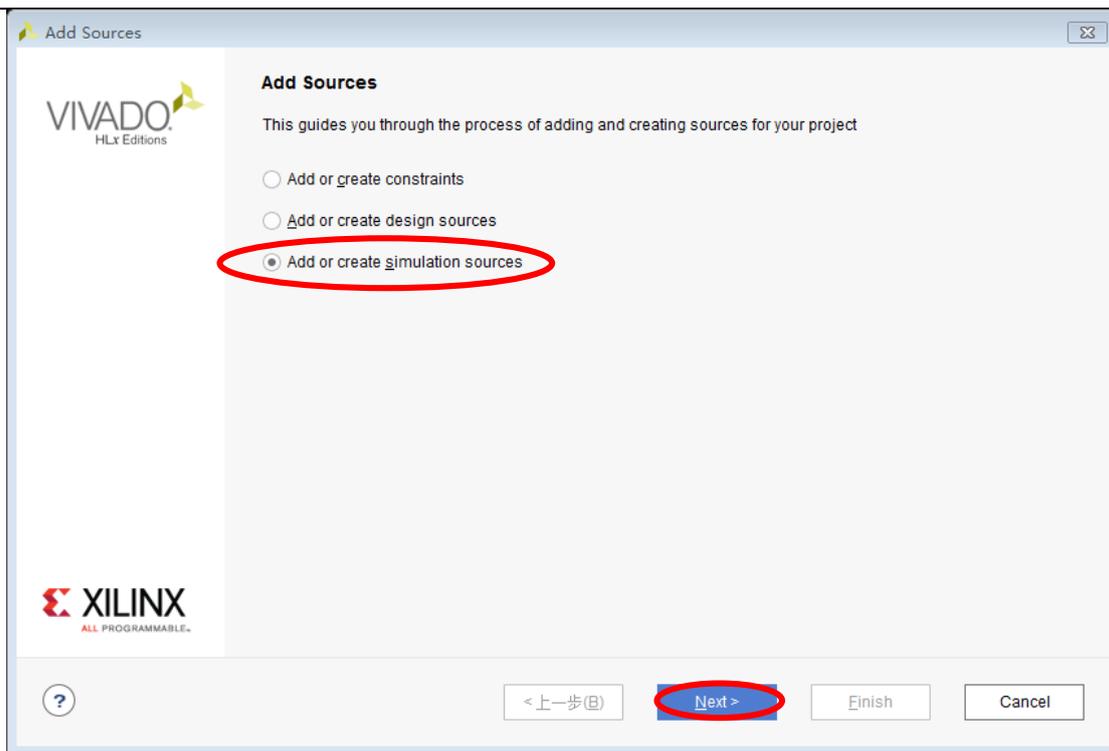


图 1-17 添加和新建测试文件

选择“Create File”，新建一个激励测试文件。

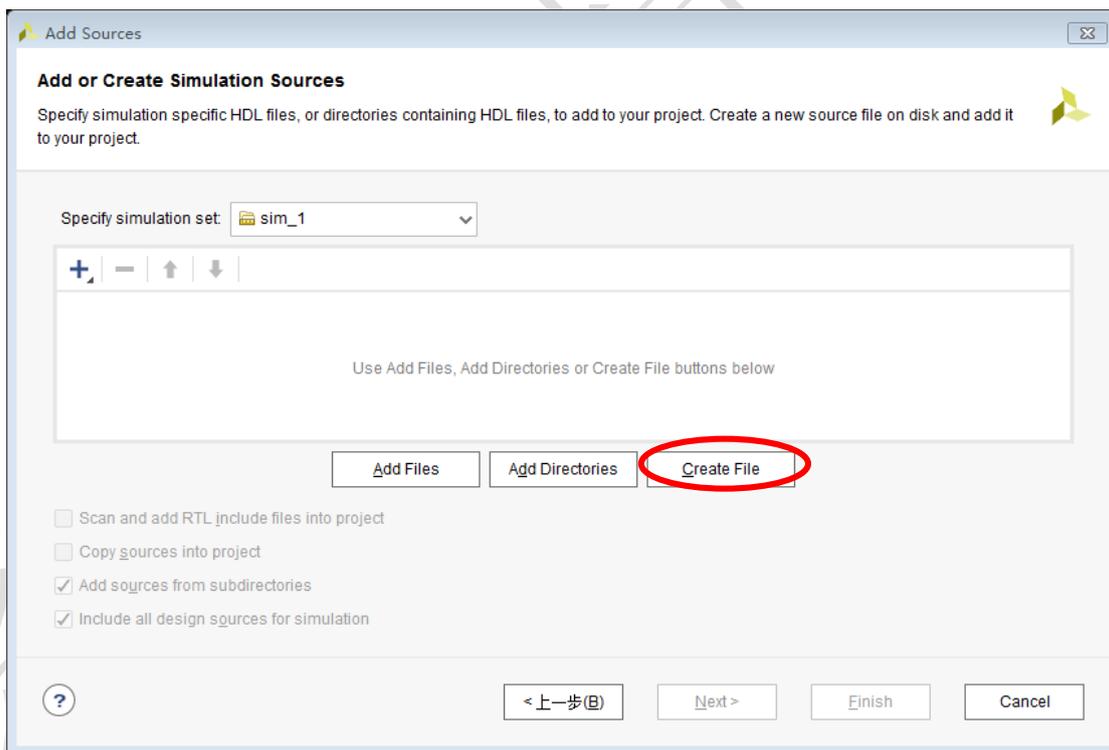


图 1-18 新建测试文件

输入激励测试文件名，点击“OK”。

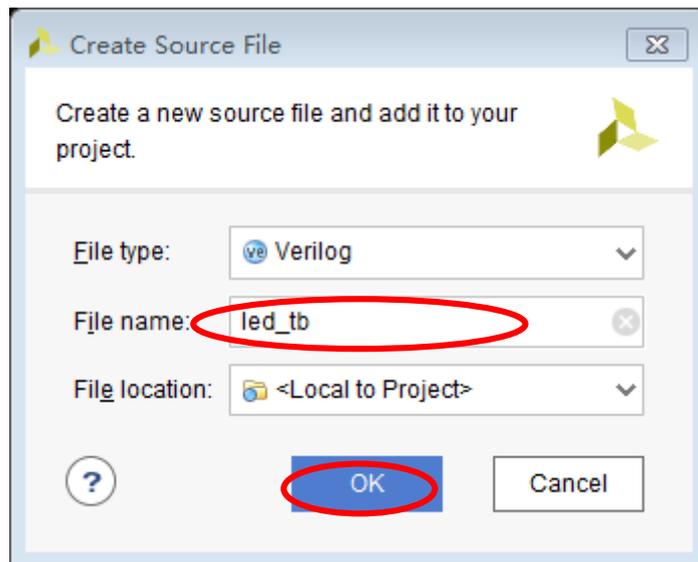


图 1-19 新建测试文件设置

完成新建测试文件，点击“Finish”。

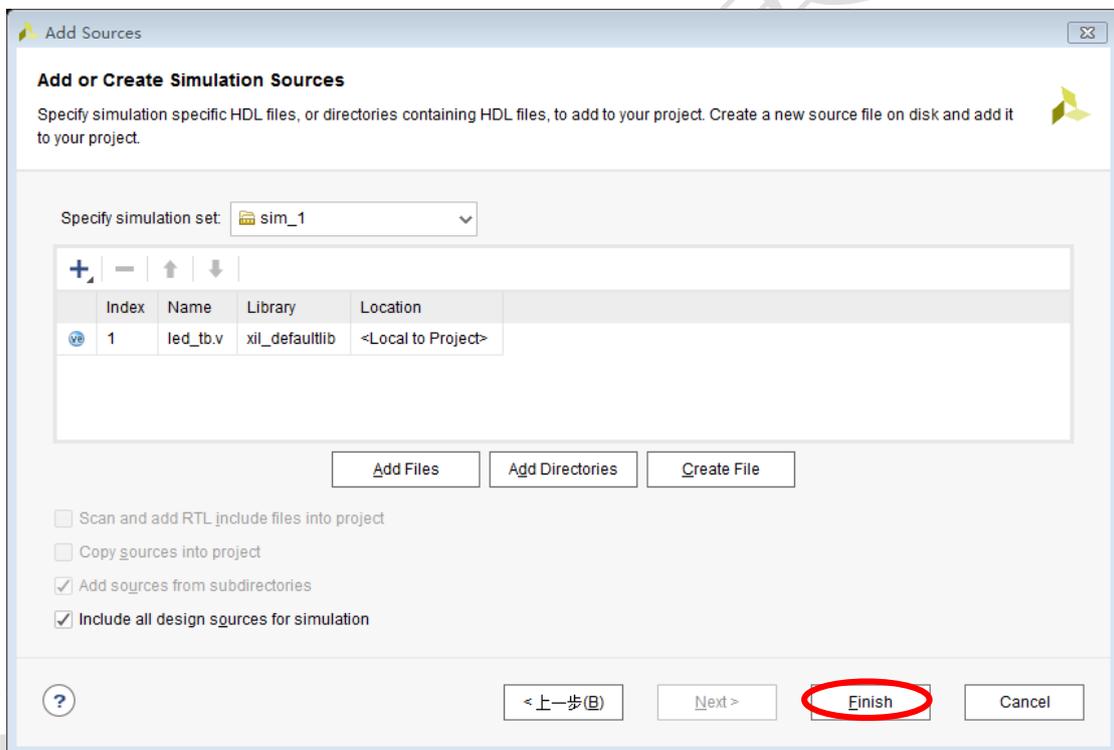


图 1-20 完成新建测试文件

对测试激励文件进行 module 端口定义，由于激励测试文件不需要有对外的接口，所以不进行 I/O 端口设置直接点击“OK”，完成空白的激励测试文件创建。

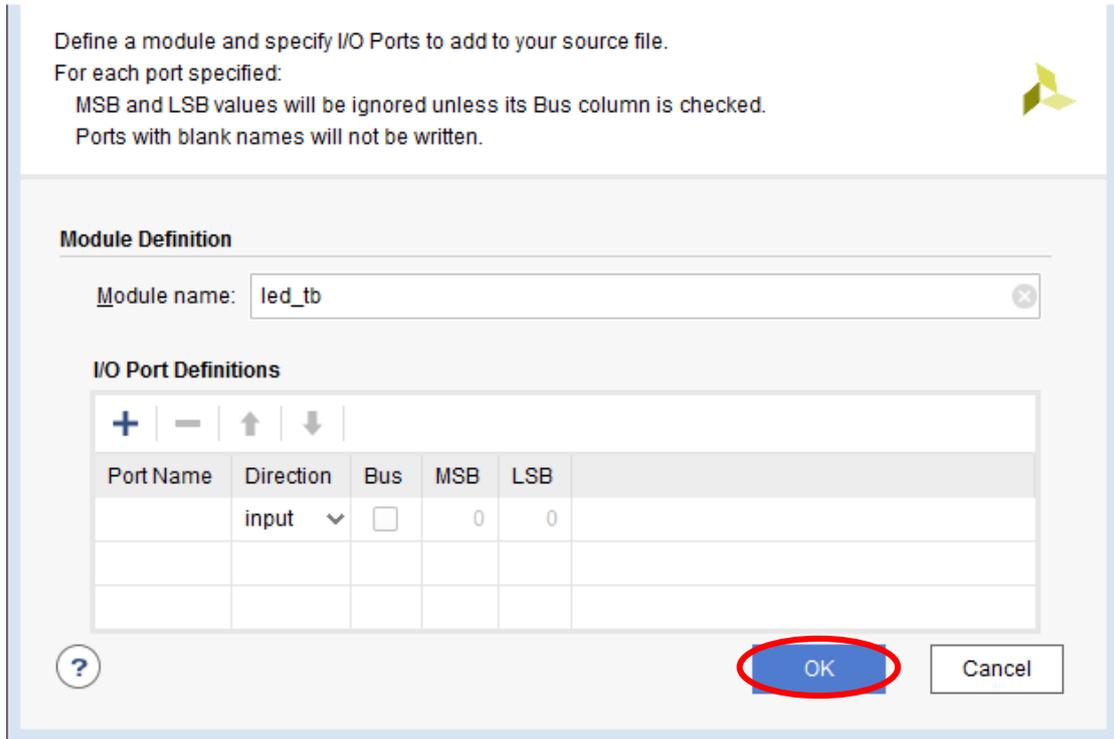


图 1-21 不需要进行端口设置

在“Source”窗口下双击打开空白的激励测试文件。测试文件 led_tb.v 位于工程目录下“\led.srcs\sim_1\new”文件夹下。

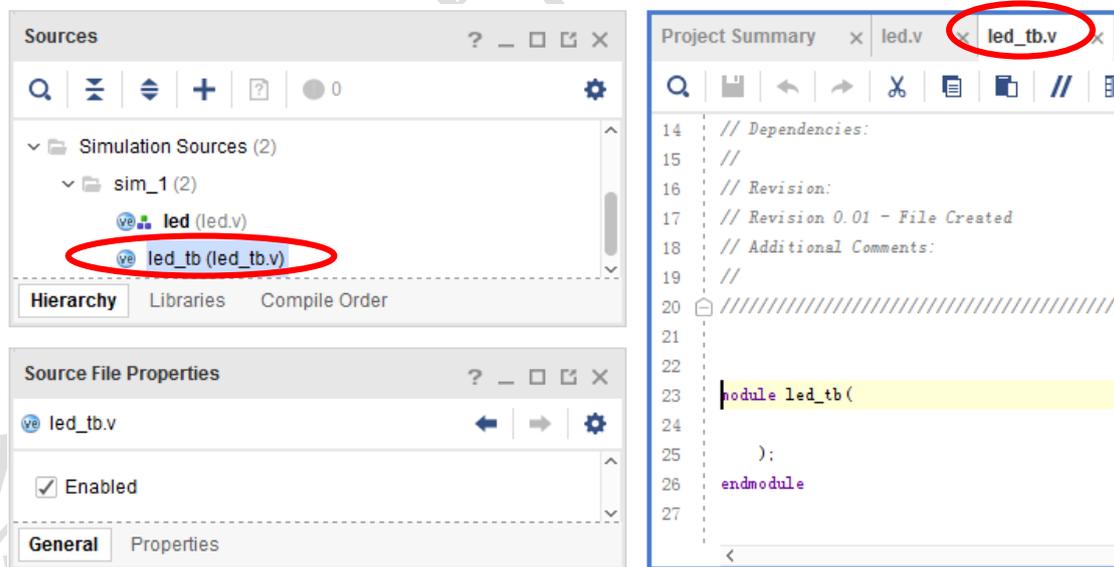


图 1-22 空白测试激励文件

完成对将要仿真的 module 的实例化和激励代码的编写，如下述代码所示。

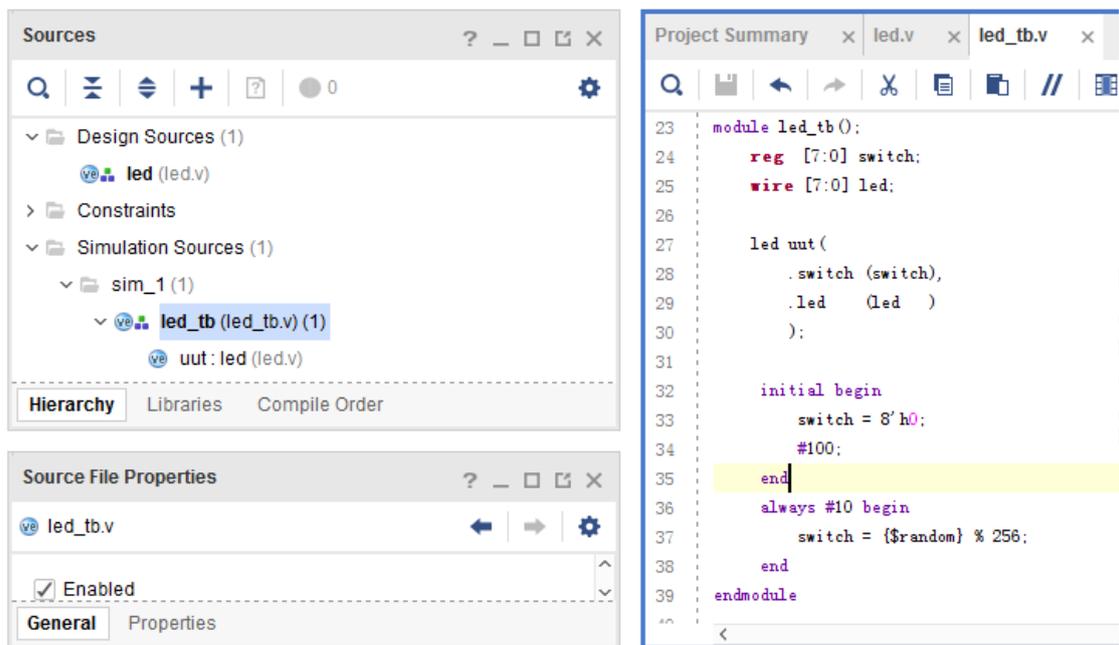


图 1-23 完成测试激励文件

进入仿真。在左侧“Flow Navigator”窗口中点击“Simulation”下的“Run Simulation”选项，选择“Run Behavioral Simulation”，进入仿真界面。

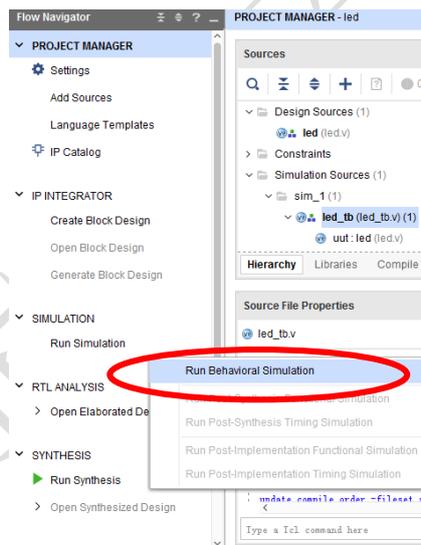


图 1-24 运行行为级仿真

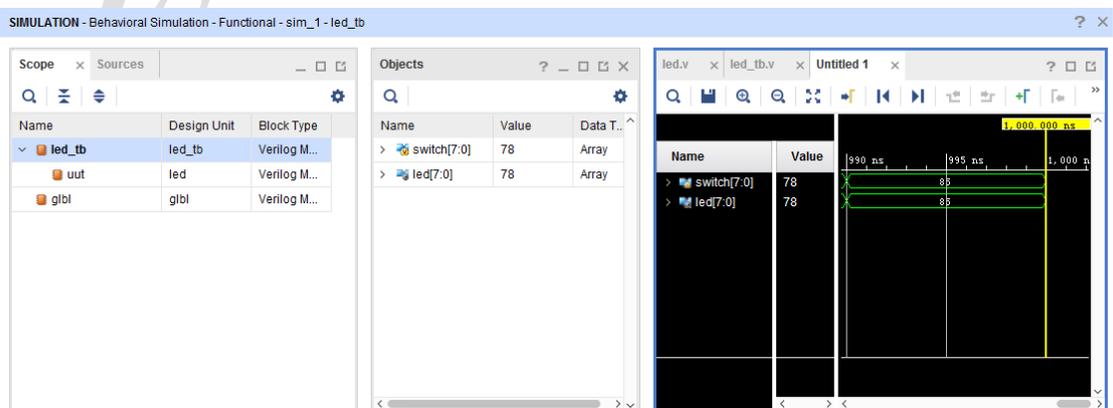


图 1-25 行为级仿真界面

可通过左侧“Scope”一栏中的目录结构定位到想要查看的 module 内部信号，在“Objects”对应的信号名称上右击选择“Add To Wave Window”，将信号加入波形图中。仿真器默认显示 I/O 信号，由于这个示例不存在内部信号，因此不需要添加观察信号。

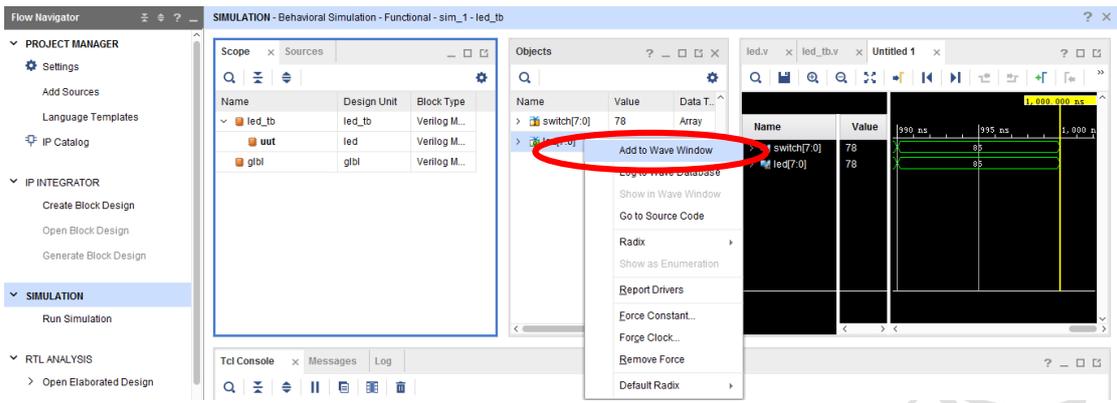


图 1-26 将内部信号添加到波形图

可通过选择工具栏中的选项来进行波形的仿真时间控制。如下图工具条，分别是复位波形（即清空现有波形）、运行仿真、运行特定时长的仿真、仿真时长设置、仿真时长单位、单步运行、暂停、重新启动。



图 1-27 仿真控制工具

观察仿真波形是否符合预期功能。在波形显示窗口上侧是波形图控制工具，由左到右分别是：查找、保存波形配置、放大、缩小、缩放到全显示、缩放到光标、转到时间 0、转到时间的最后、前一个跳变、下一次跳变、添加标记、前标记、下一个标记、交换光标。



图 1-28 波形图控制工具

可通过右键选中信号来改变信号的显示形态。如下图将信号改为二进制显示。

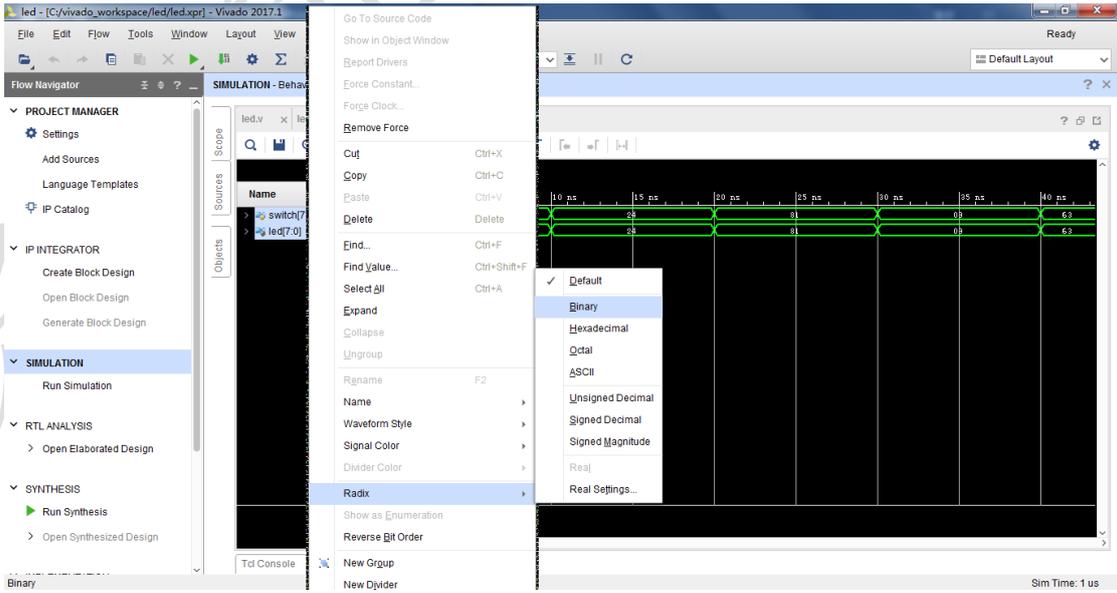


图 1-29 仿真波形窗口

查看波形检查设计功能正确性。

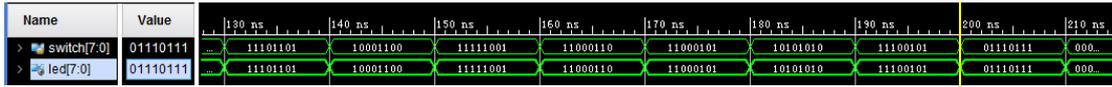


图 1-30 仿真波形结果

1.5 添加约束文件

添加约束文件有两种方法，一是利用 Vivado 中 I/O Planning 功能，二是直接新建 XDC 的约束文件，手动输入约束命令。下面分别介绍这两种方法：

1.5.1 利用 I/O Planning 生成约束文件

点击“Flow Navigator”中“Synthesis”下的“Run Synthesis”，在弹出的“Launch Runs”窗口点“OK”，先对工程进行综合。

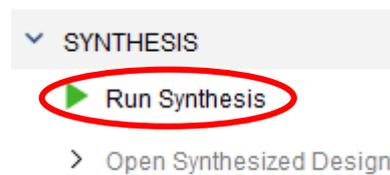


图 1-31 综合工程

综合完成之后，选择“Open Synthesized Design”，打开综合后的网表。

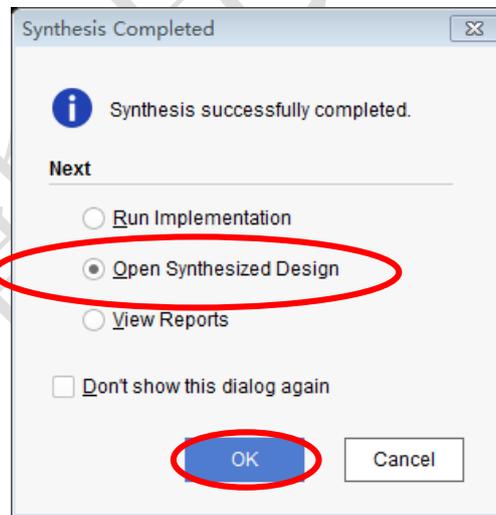


图 1-32 打开综合结果

此时应看到如下界面，如果没出现如下界面，在图示位置的“layout”中选择“I/O Planning”一项。

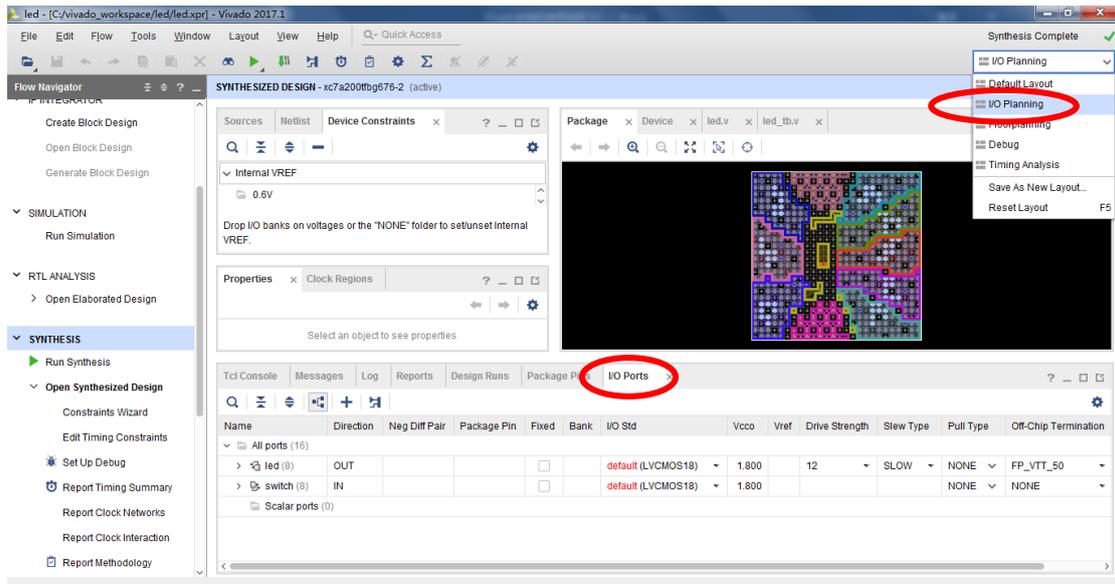


图 1-33 I/O Planning 界面

在界面右下方的选项卡中切换到“IO Ports”一栏，并在对应的信号后，在“Site”一列输入对应的 FPGA 管脚标号（或将信号拖拽到右上方“Package”图中对应的管脚上），并指定“I/O std”。具体的 FPGA 约束管脚和 I/O 电平标准，可参考对应板卡的原理图或提供的常用的引脚对应关系表（参见 龙芯 Artix-7 实验箱-原理图与引脚列表）。

	A	B	C
1			引脚对应关系
2	引脚名称	FPGA接口编号	描述
3	LED灯		
4	FPGA_LED1	H7	实验板放正，一排LED灯左起第一个。
5	FPGA_LED2	D5	依次类推
6	FPGA_LED3	A3	
7	FPGA_LED4	A5	
8	FPGA_LED5	A4	
9	FPGA_LED6	F7	
10	FPGA_LED7	G8	
11	FPGA_LED8	H8	
12	FPGA_LED9	J8	
13	FPGA_LED10	J23	
14	FPGA_LED11	J26	
15	FPGA_LED12	G9	
16	FPGA_LED13	J19	
17	FPGA_LED14	H23	
18	FPGA_LED15	J21	
19	FPGA_LED16	K23	实验板放正，一排LED灯左起第八个。

图 1-34 引脚对应关系表

完成 I/O Ports 设置如下图，“I/O Std”设置为“LVCMOS33*”表明 I/O 标准是 3.3V 的 LVCMOS。

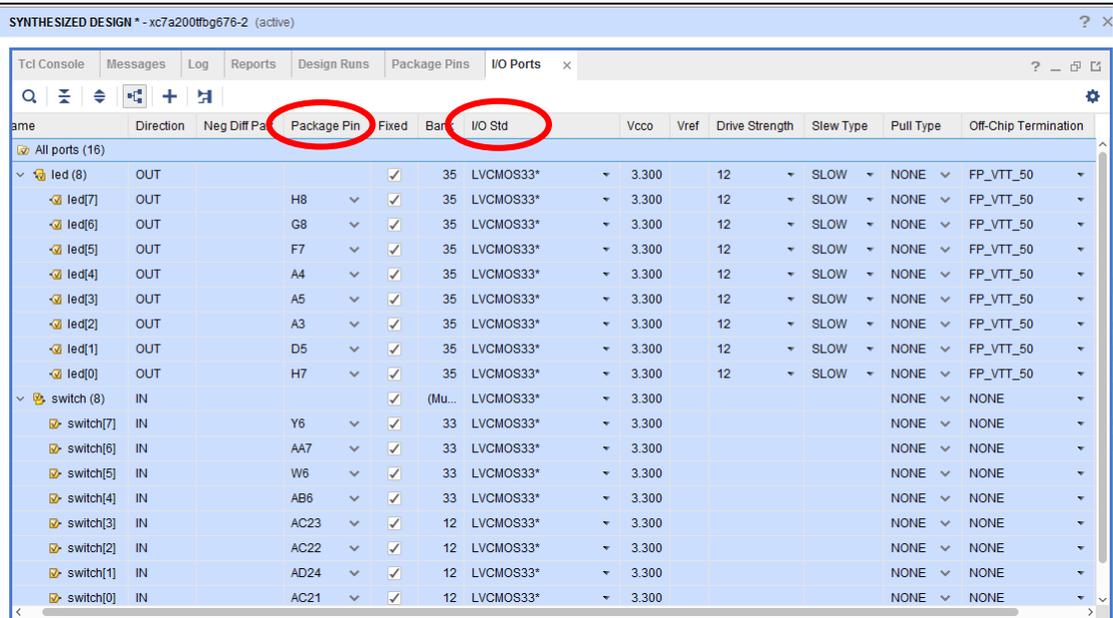


图 1-35 完成 I/O 端口设置

点击左上方工具栏中的保存按钮，提示添加新的约束文件使综合过时失效。点击“OK”。

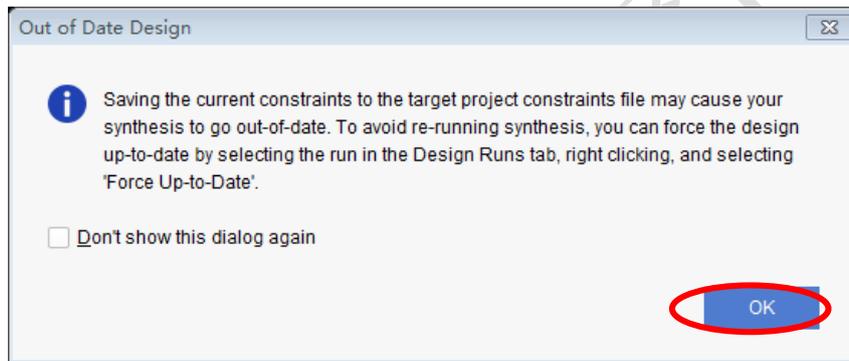


图 1-36 提示综合失效

新建 XDC 文件或选择工程中已有的 XDC 文件。选择“Create a new file”，输入“File name”，点击“OK”完成约束过程。

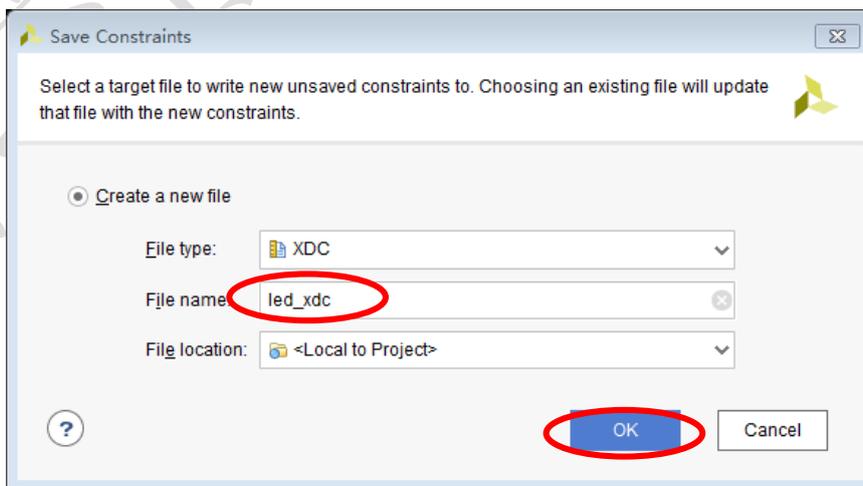


图 1-37 新建 XDC 文件设置

在“Sources”窗口“Constraints”层次下可以查看新建的 XDC 文件。

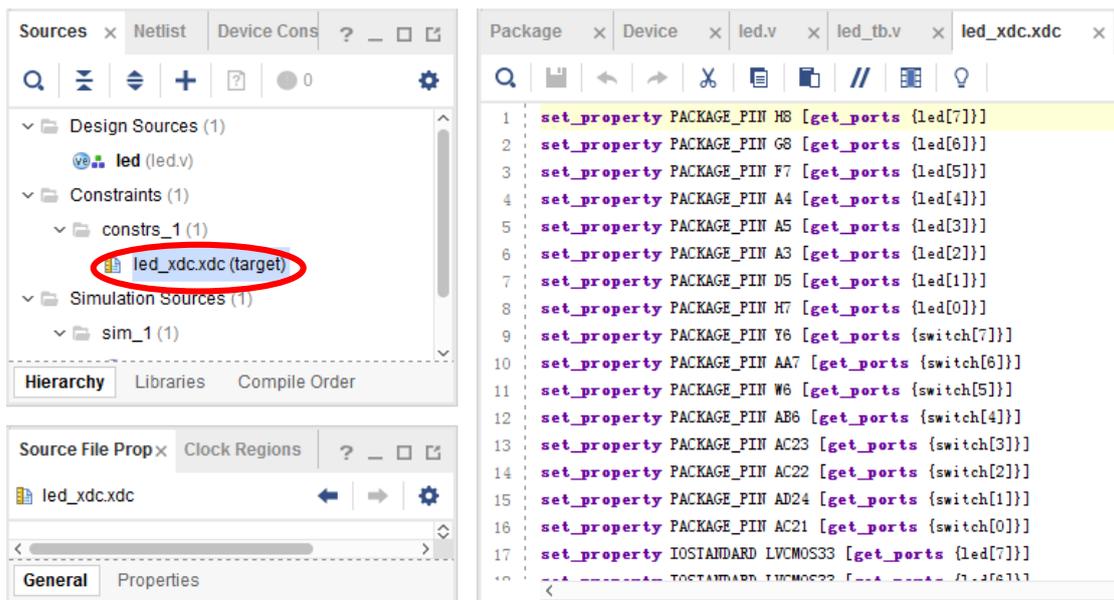


图 1-38 完成 XDC 文件创建

1.5.2 新建 XDC 文件并手动输入约束

新建约束文件。点击“Add Sources”，选择“Add or Create Constraints”，点击“Next”。

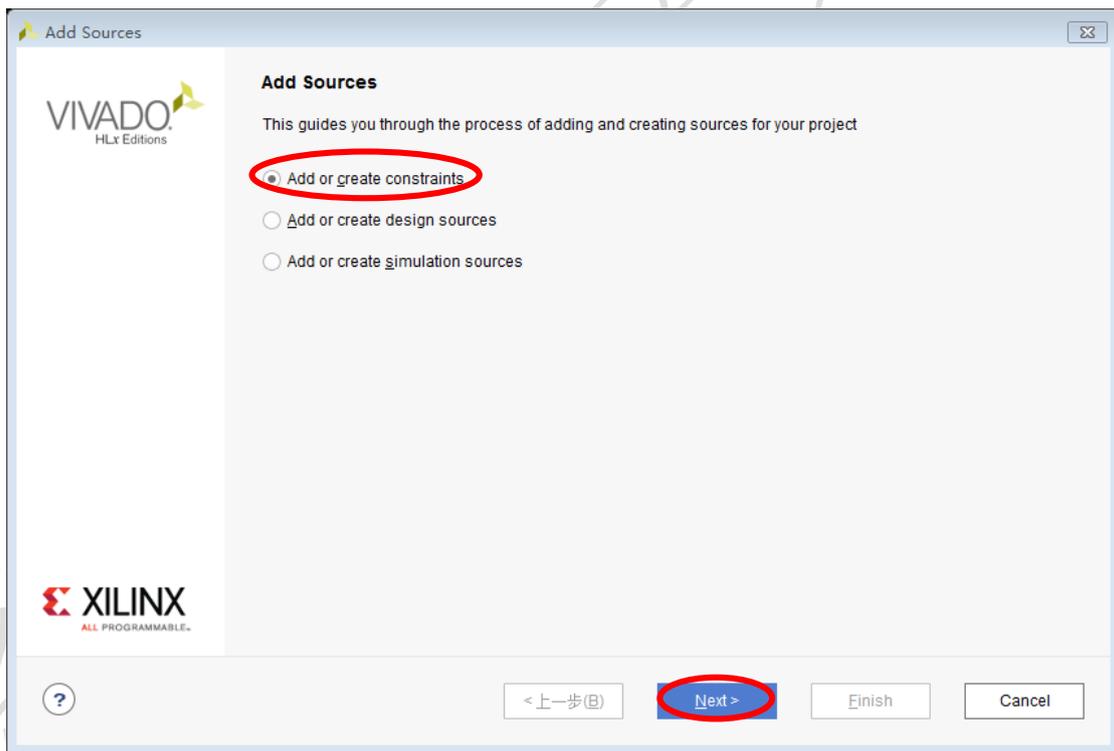


图 1-39 新建约束文件

添加或创建约束文件。点击“Create Files”添加约束文件。

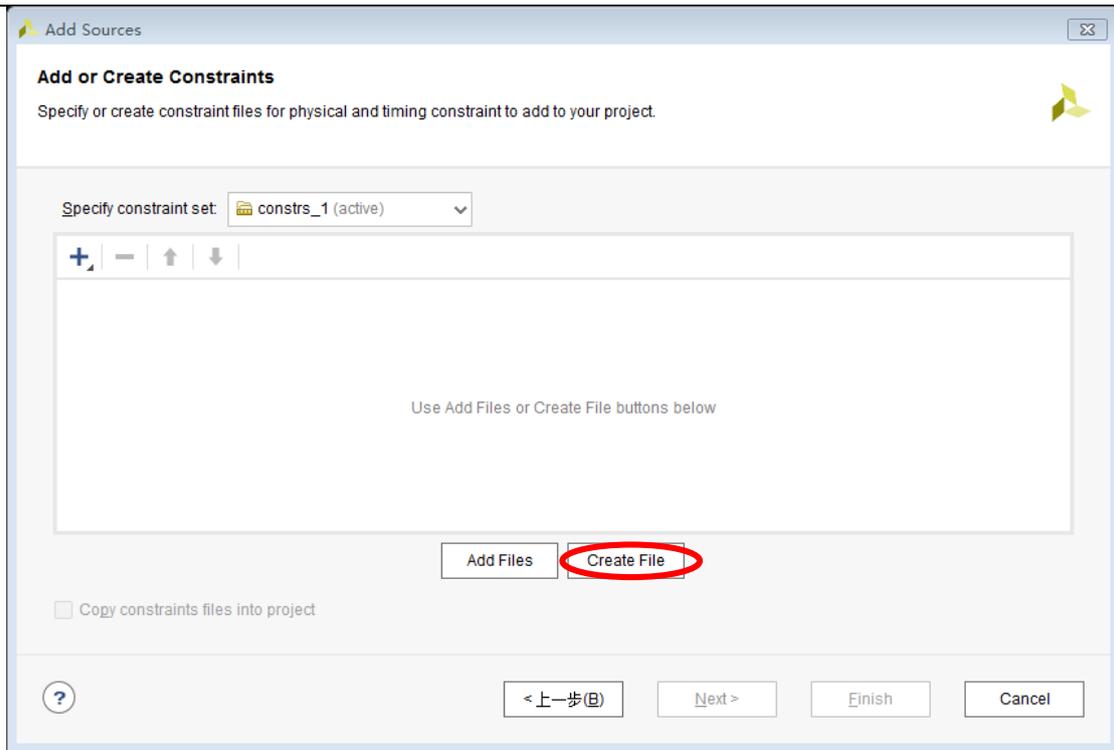


图 1-40 添加或创建约束文件

设置新建的 XDC 文件，输入 XDC 文件名，点击“OK”。默认文件在工程目录下的“\led.srcs\constrs_1\new”中。

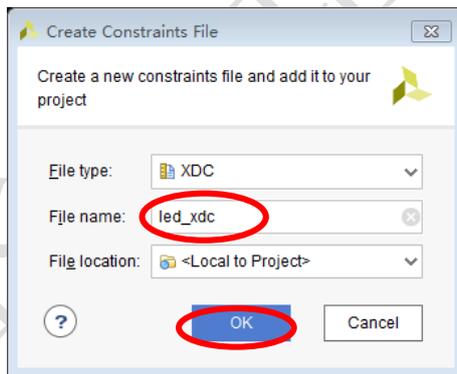


图 1-41 新建 XDC 文件设置

完成新建 XDC 文件，点击“Finish”。

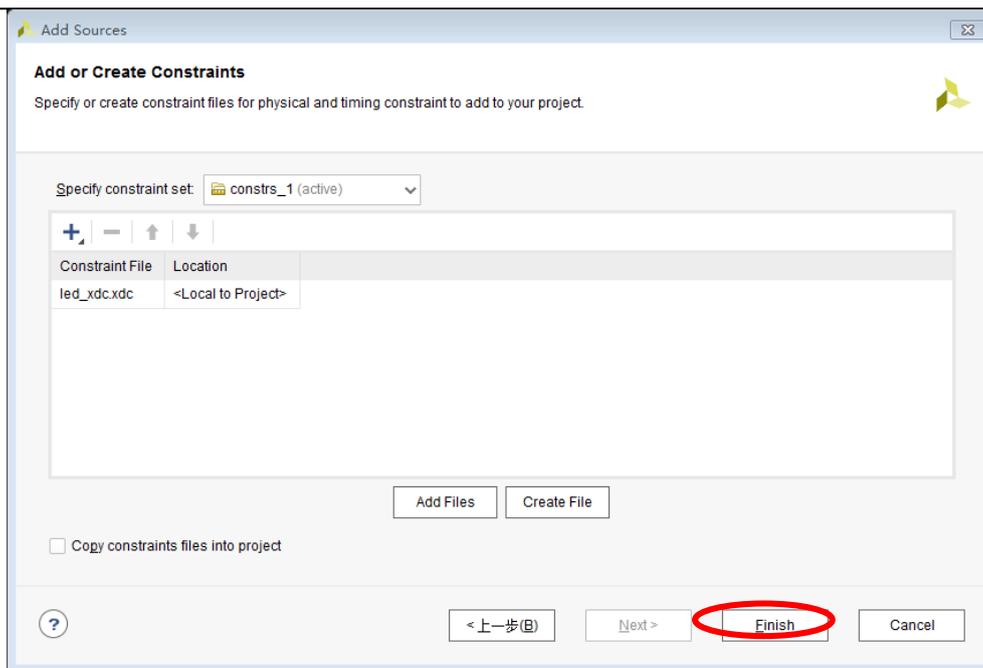


图 1-42 完成新建 XDC 文件

在“Sources”窗口“Constraints”下双击打开新建好的 XDC 文件“led_xdc.xdc”，并参照下面的约束文件格式，输入相应的 FPGA 管脚约束信息和电平标准。

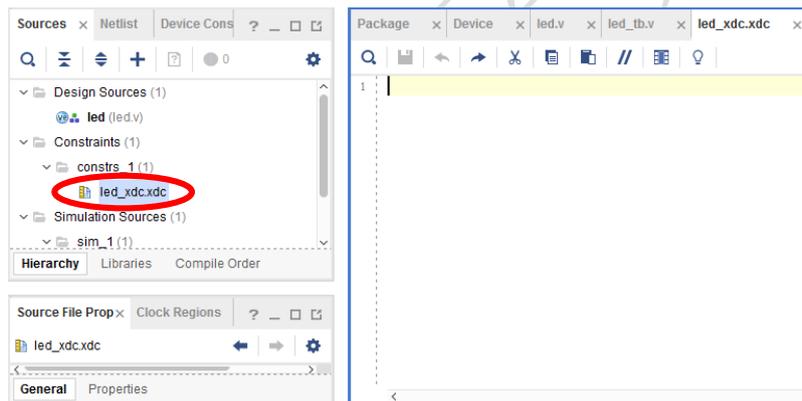


图 1-43 打开空白 XDC 文件

约束文件参考：

```

1 set_property PACKAGE_PIN H8 [get_ports {led[7]}]
2 set_property PACKAGE_PIN G8 [get_ports {led[6]}]
3 set_property PACKAGE_PIN F7 [get_ports {led[5]}]
4 set_property PACKAGE_PIN A4 [get_ports {led[4]}]
5 set_property PACKAGE_PIN A5 [get_ports {led[3]}]
6 set_property PACKAGE_PIN A3 [get_ports {led[2]}]
7 set_property PACKAGE_PIN D5 [get_ports {led[1]}]
8 set_property PACKAGE_PIN H7 [get_ports {led[0]}]
9 set_property PACKAGE_PIN Y6 [get_ports {switch[7]}]
10 set_property PACKAGE_PIN AA7 [get_ports {switch[6]}]
11 set_property PACKAGE_PIN W6 [get_ports {switch[5]}]
12 set_property PACKAGE_PIN AB6 [get_ports {switch[4]}]
13 set_property PACKAGE_PIN AC23 [get_ports {switch[3]}]
14 set_property PACKAGE_PIN AC22 [get_ports {switch[2]}]

```

```
15 set_property PACKAGE_PIN AD24 [get_ports {switch[1]}]
16 set_property PACKAGE_PIN AC21 [get_ports {switch[0]}]
17 set_property IOSTANDARD LVCMOS33 [get_ports {led[7]}]
18 set_property IOSTANDARD LVCMOS33 [get_ports {led[6]}]
19 set_property IOSTANDARD LVCMOS33 [get_ports {led[5]}]
20 set_property IOSTANDARD LVCMOS33 [get_ports {led[4]}]
21 set_property IOSTANDARD LVCMOS33 [get_ports {led[3]}]
22 set_property IOSTANDARD LVCMOS33 [get_ports {led[2]}]
23 set_property IOSTANDARD LVCMOS33 [get_ports {led[1]}]
24 set_property IOSTANDARD LVCMOS33 [get_ports {led[0]}]
25 set_property IOSTANDARD LVCMOS33 [get_ports {switch[7]}]
26 set_property IOSTANDARD LVCMOS33 [get_ports {switch[6]}]
27 set_property IOSTANDARD LVCMOS33 [get_ports {switch[5]}]
28 set_property IOSTANDARD LVCMOS33 [get_ports {switch[4]}]
29 set_property IOSTANDARD LVCMOS33 [get_ports {switch[3]}]
30 set_property IOSTANDARD LVCMOS33 [get_ports {switch[2]}]
31 set_property IOSTANDARD LVCMOS33 [get_ports {switch[1]}]
32 set_property IOSTANDARD LVCMOS33 [get_ports {switch[0]}]
```

1.6 工程实现

在“Flow Navigator”中点击“Program and Debug”下的“Generate Bitstream”选项，工程会自动完成综合、布局布线、Bit 文件生成过程，完成之后，可点击“Open Implemented Design”来查看工程实现结果。

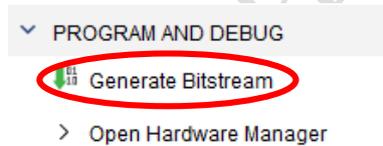


图 1-44 生成 bit 流文件

提示综合过时重新运行综合和实现，点击“**Yes**”。在弹出的“Launch Runs”窗口点“OK”。

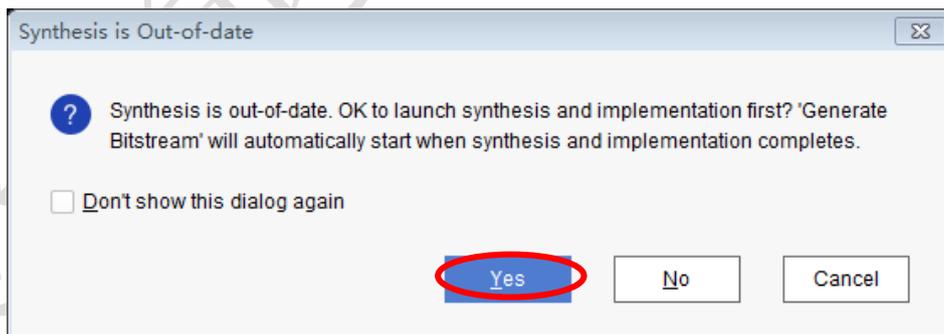


图 1-45 提示重新运行综合和实现

在比特流文件生成完成的窗口选择“Open Hardware Manager”，进入硬件管理界面。连接 FPGA 开发板的电源线和与电脑的下载线，打开 FPGA 电源。

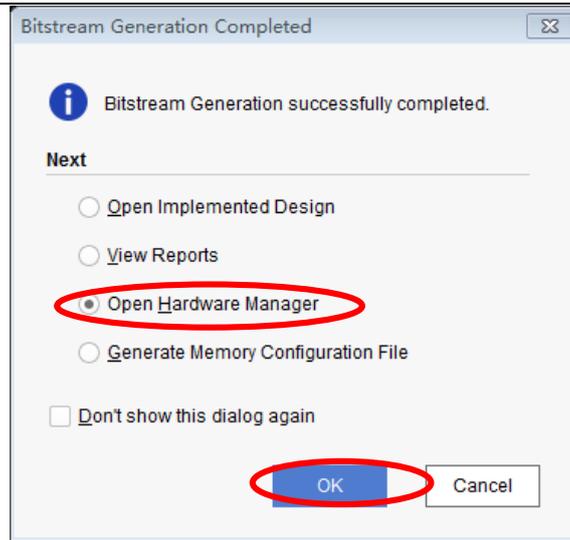


图 1-46 打开硬件程序和调试管理

在“Hardware Manager”窗口的提示信息中，点击“Open target”的下拉菜单的“Open New Target”（或在“Flow Navigator”下“Program and Debug”中展开“Open Hardware Manager”，点击“Open Target”->“Open New Target”）。也可以选择“Auto Connect”自动连接器件。

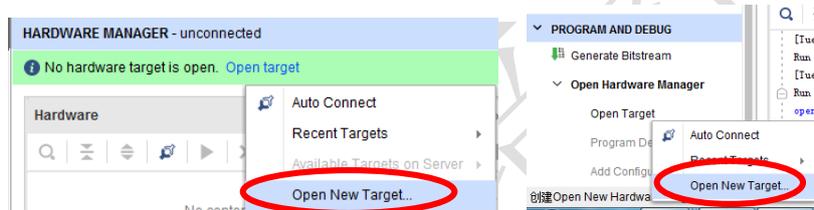


图 1-47 打开新目标

在“Open Hardware Target”向导中，先点击“Next”，进入 Server 选择向导。

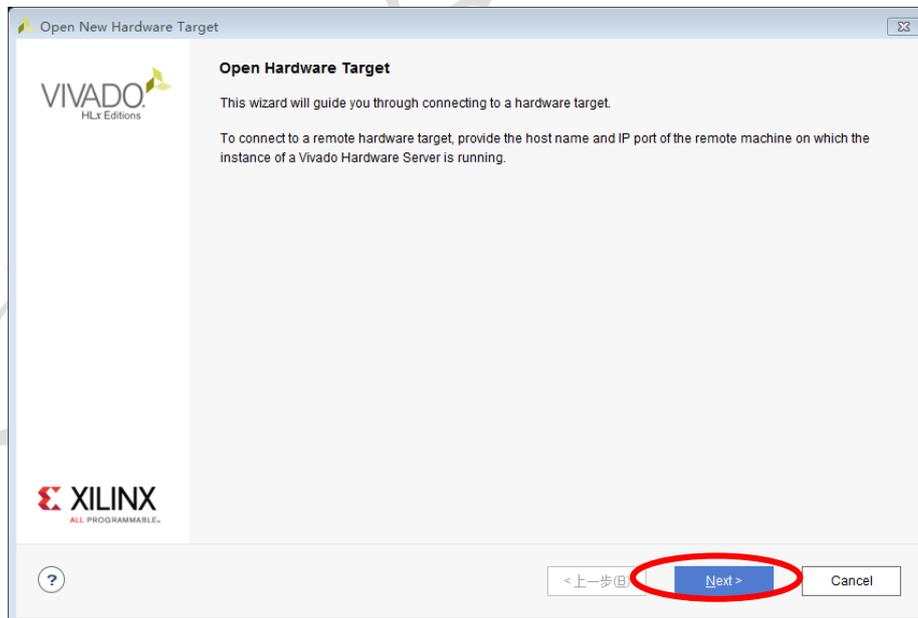


图 1-48 Open Hardware Target 向导

选择连接到“Local server”，点击“Next”。

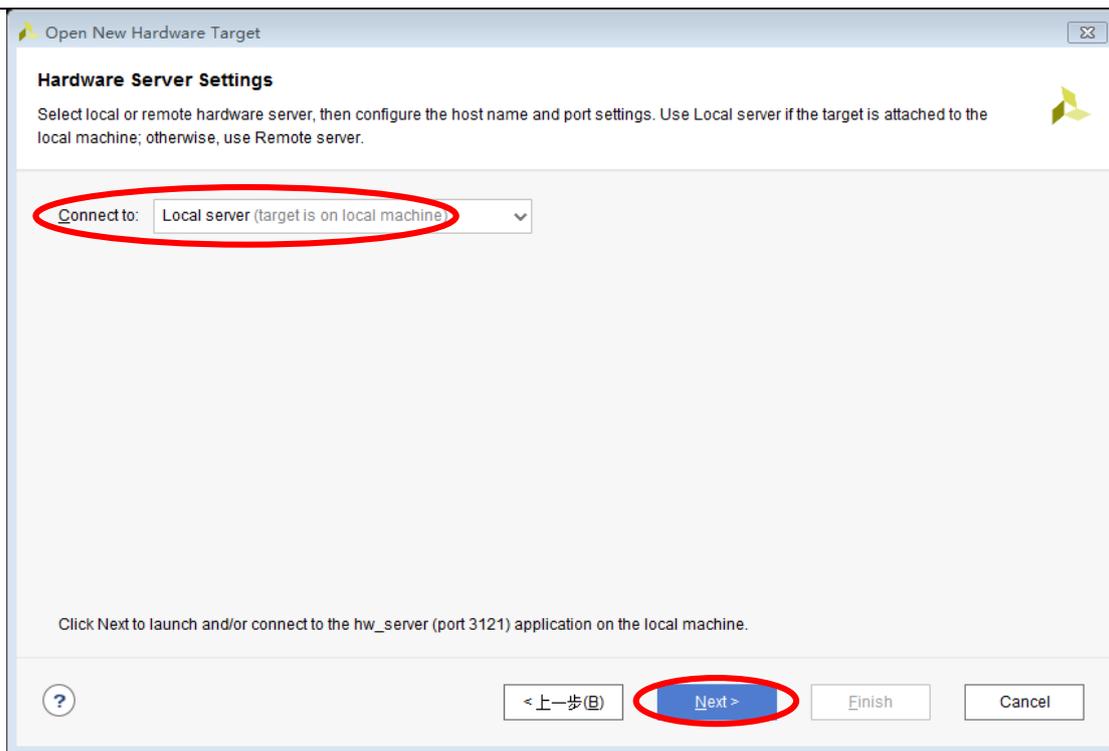


图 1-49 连接 Local server

选择目标硬件，点击“Next”。

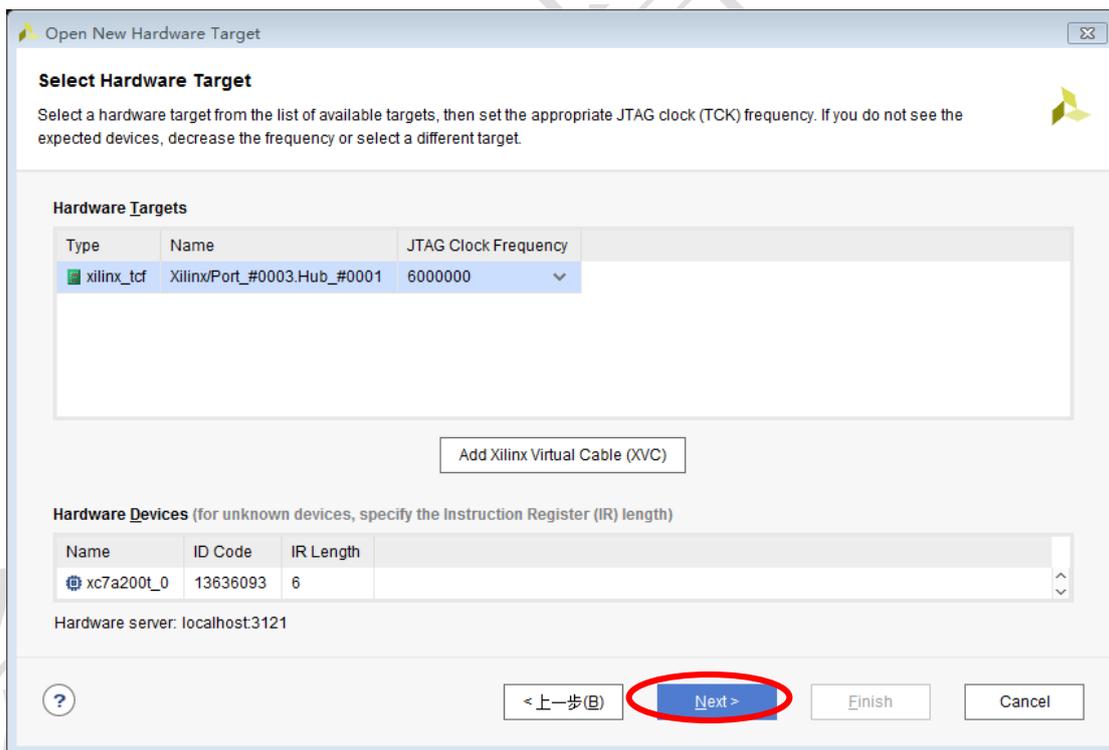


图 1-50 选择目标硬件

完成目标硬件打开。点击“Finish”。

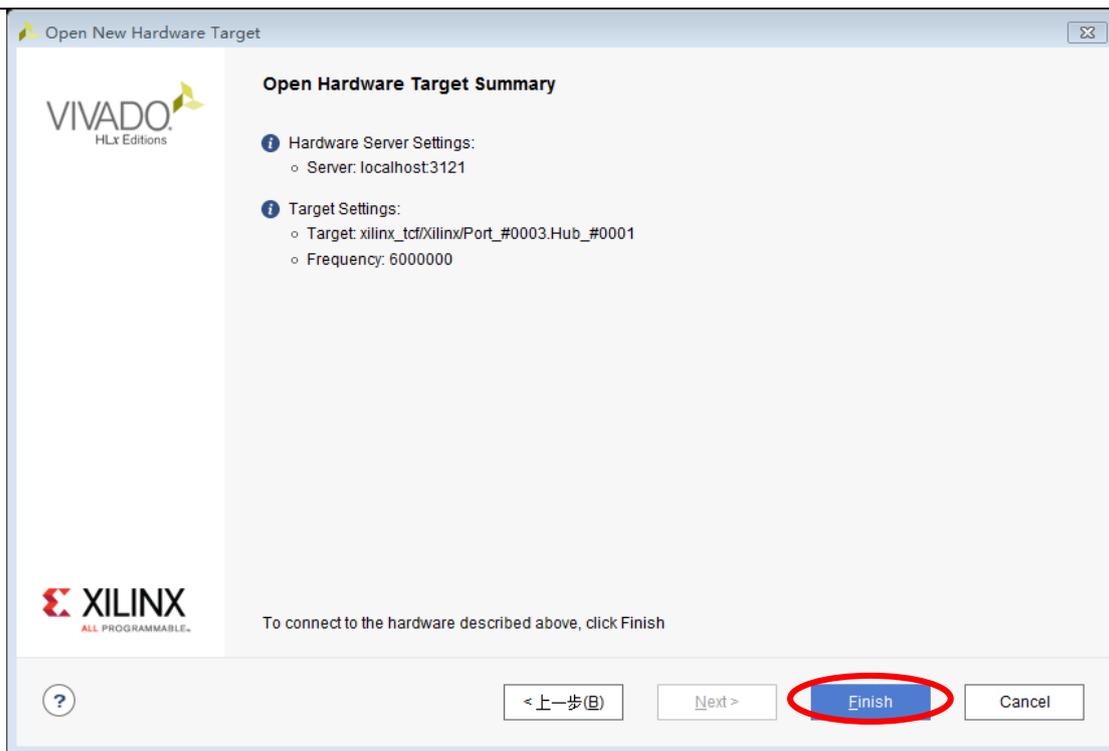


图 1-51 完成目标硬件打开

对目标硬件编程。在“Hardware”窗口右键单击目标器件“xc7a200t_0”，选择“Program Device...”。或者“Flow Navigator”窗口中“Program and Debug”->“Hardware Manager”->“Program Device”。

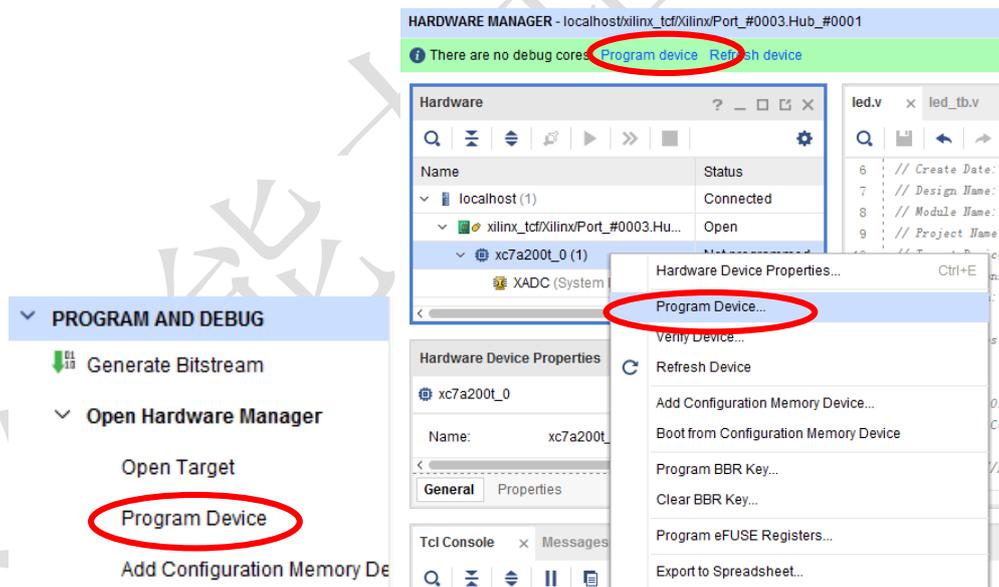


图 1-52 对目标器件编程

选择下载的 bit 流文件，点击“Program”。

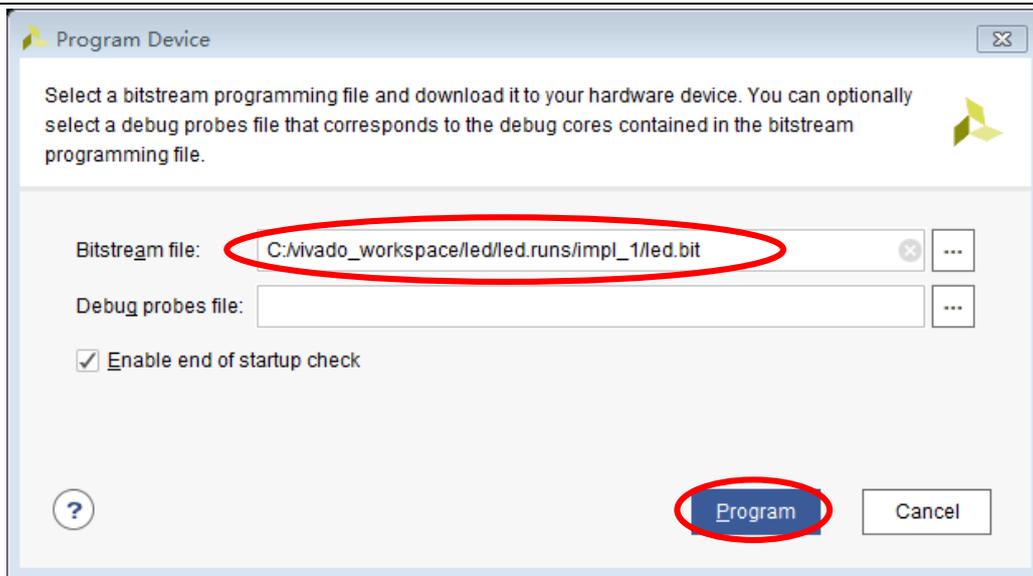


图 1-53 选择 bit 流文件

完成下载后，“Hardware”窗口下的“xc7a200t_0”的状态变成“Programmed”。

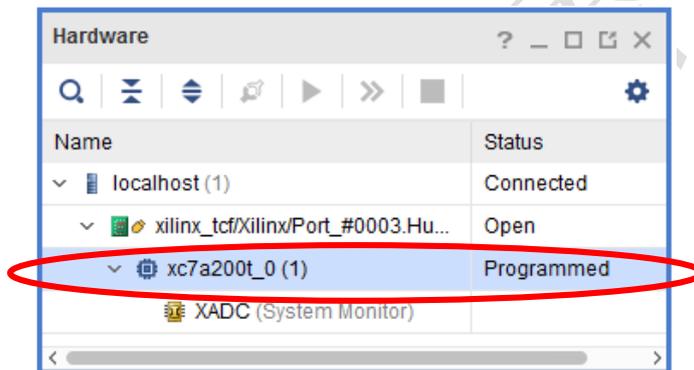


图 1-54 完成 FPGA 编程

FPGA 开发板上，8 个拨码开关 SW18~SW25 对应控制 8 个 led 灯 LED1~LED8 的亮灭。设计完成。

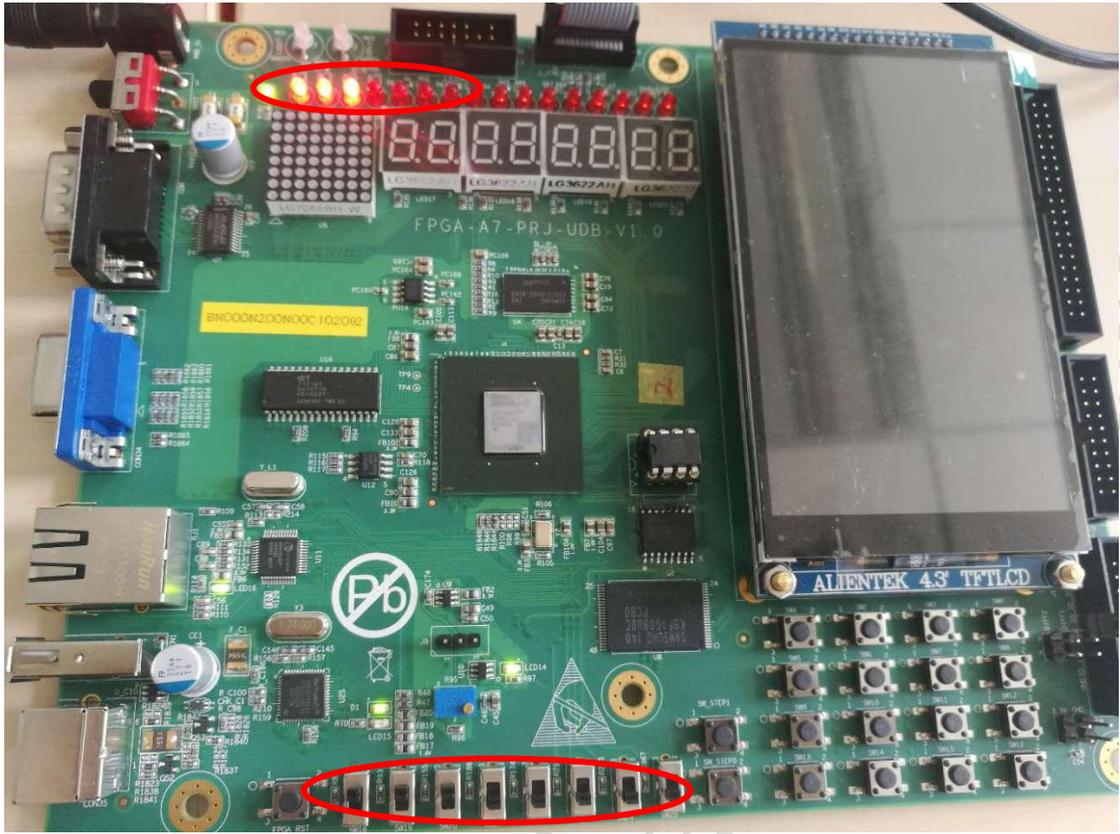


图 1-55 FPGA 运行结果