

计算机系统-CPU实验

关于同学们能学到什么

- 更加深刻、全面地理解老师课堂上所讲的计算机系统结构知识、CPU的工作过程。相信大家现在对这门课的难度和在它cs专业中的重要性有了一定认知，认真做好实验，不光会锻炼能力，甚至能让你们的期末复习变得轻松。
- 入门一种新的编程语言：`verilog`。大家可能会觉得搞AI的会 `python`、`cuda` 就很了不起了，但是多学点总没坏处。
- 掌握 `Linux` 平台以及 `Git` 工具的基础操作。这也会对大家以后的学习、科研或者工作有很大帮助。
- 我们的实验内容基于“龙芯杯”全国大学生系统能力培养大赛，在实验中如果有感兴趣的同学也欢迎联系我，后续有机会可以参加明年的比赛。

关于基础需要会什么

考虑到大家的课业安排，我尽可能让大家好上手，以实践课堂所学的理论为主。

实验环境（所需软件和项目包）已经给大家在CG平台准备好，该平台后续也会供你们做Deep Learning相关的项目使用，大家今年有机会提前上手。

`verilog` 作为一门类C语言，相信对大家来说应该不是问题。在实验项目中有部分知识需要大家查阅书籍，相信大三的同学应该也具备不错的自学能力了。

总的来说，实验项目不难，希望大家能够安排好时间，认真对待。

关于考核

实验建议2-3人一组，小组评分标准如下（组员分数会以小组分数为基准进行浮动）：

1.功能分 60分（通过 `func_test`）

```
10分 pass point 1
10分 pass point 36
10分 pass point 43
10分 pass point 51
10分 pass point 58
10分 pass point 64
```

2.验收分40分（具体分配方式可能会根据同学们的实验情况进行调整）

```
15分 报告
10分 现场测试（加、改代码）
10分 可选模块
5分  github仓库记录
```

2.1 可选模块

1. 将CPU封装为AXI接口，func_test通过P'64测试点
2. 通过P'89测试点
3. 通过性能测试，且能够上板验证
4. 改进流水线结构，例如六级及以上的流水段，多发射等
5. 其它内容（提前联系助教进行评估）

还是那句话，实验课程不是为了难为大家，希望大家做完实验后这门课能学得更好。虽然现在看起来任务量不少，但是大家动手认真做就能获得不错的成绩。

如果有同学主动把培养能力的机会全部交给队友，那助教也会在验收时酌情让他的成绩多浮动一些.....（助教脾气好，但是不好骗，这项要求是底线）

关于后续安排

虽然课表安排的实验课在十二月份，但是考虑到项目工作量，希望大家可以今早开始，为期末周的自己多着想。

11.5之前要做的事：

- 组队，和队友协商分工
- 登录自己的CG平台，熟悉服务器环境
- 准备团队的GitHub仓库，学习git基础
- 入门verilog语言，完成一个小任务

Tips:

1.CG平台服务器的使用

同学们登录自己的CG平台，在课程里面找到 计算机系统 -> 在线实验 -> 任务一 开始实验 -> 带有GUI的服务器 界面。

点击 更多 里面可以选择交互操作，注意服务器装不了中文，每天0点时除了 /mnt/cgshare 目录外，其它位置会恢复为初始镜像状态。因此建议大家在工作结束后及时将代码上传或是下载到自己的电脑。

重新连接桌面	和浏览器刷新页面一样，如果分辨率卡死或者桌面卡住可以先考虑这个方法
剪切板	和本地电脑的剪切操作都需要通过这个中转
下载文件	下载 cgshare 目录下的文件，文件夹要打压缩包，强烈建议把代码在本地存一份
上传文件	上传内容到 cgshare 目录，小于128M，文件夹要打压缩包
重新启动	和电脑的重启一样，没保存的代码修改会丢失，但是不会清理文件
桌面还原	还原到初始镜像状态，慎点，除非把文件系统搞坏了

软件环境已经准备好，还需要相关的Linux操作大家自行百度。遇见服务器问题解决不了私聊我就好。

2.Git仓库管理

这里推荐大家用 `GitHub`，也是以后工作流中的主力，访问方法和创建账户大家自行解决，同学们可以互帮互助一下，要求每个同学都有自己的账户。每个组以 `organization` 的形式搭建自己的实验仓库，或者在组长的仓库维护自己的分支，虽然需要一定的学习成本，但是习惯了以后要比你们 qq 压缩包日期命名发来发去方便得多。里面的 `commit` 记录也会一定程度影响大家的验收分数。（[关于git命令点我看教程](#)）

大家在初次使用时很容易切错 `branch` 而导致一天白干，所以建议大家代码在本地保存一份（别放在 `git` 的目录下）。

3.Verilog语言入门

大家已经有 `c`、`c++` 的基础，参看这篇[教程](#)就可以快速入门。

我们使用的仿真环境为 `Vivado 2019.2`，简单教程放在附录。

服务器上为大家预装了 `VSCode` 以及 `Verilog` 相关的插件，如果语法有错误，在 `Ctrl+S` 后会红线高亮提示，解决完语法问题再开始仿真。

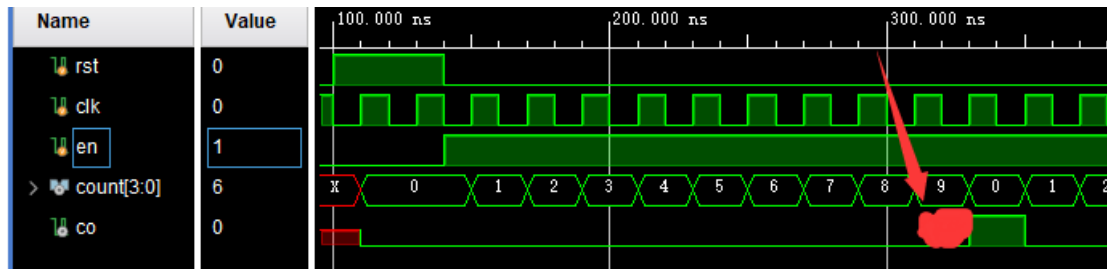
下面是一个小任务，需要以小组为单位，11月5号之前发给助教验收。有不懂的欢迎随时私聊助教，也可以在群里讨论，这样有助于你们后面的工作。

任务程序的Github地址：<https://github.com/polarispw/22-fall-CPUsample.git>

作业说明

1.修改模 10 计数器，使进位信号产生在最大那个数字的周期。

即，如果数字是 0-9，则进位信号出现在 9 那个数字出现的周期。



本次提供的示例中进位信号错位，需要同学们自行调整。箭头标出处为正确位置。

2.根据模 10 计数器，自行完成模 6 计数器。

3.级联模 10 计数器和模 6 计数器，生成模 60 计数器。

要求：模 60 计数器的进位信号每 60 个时钟周期只产生一次。

验收截图要求：如上图展示三个计数器的波形，以及提供源码截图即可。

截止时间：11 月 5 日 22 点前。

参考资料：自行百度

示例文件说明：tb 写的较为简单，可自行调整。

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。

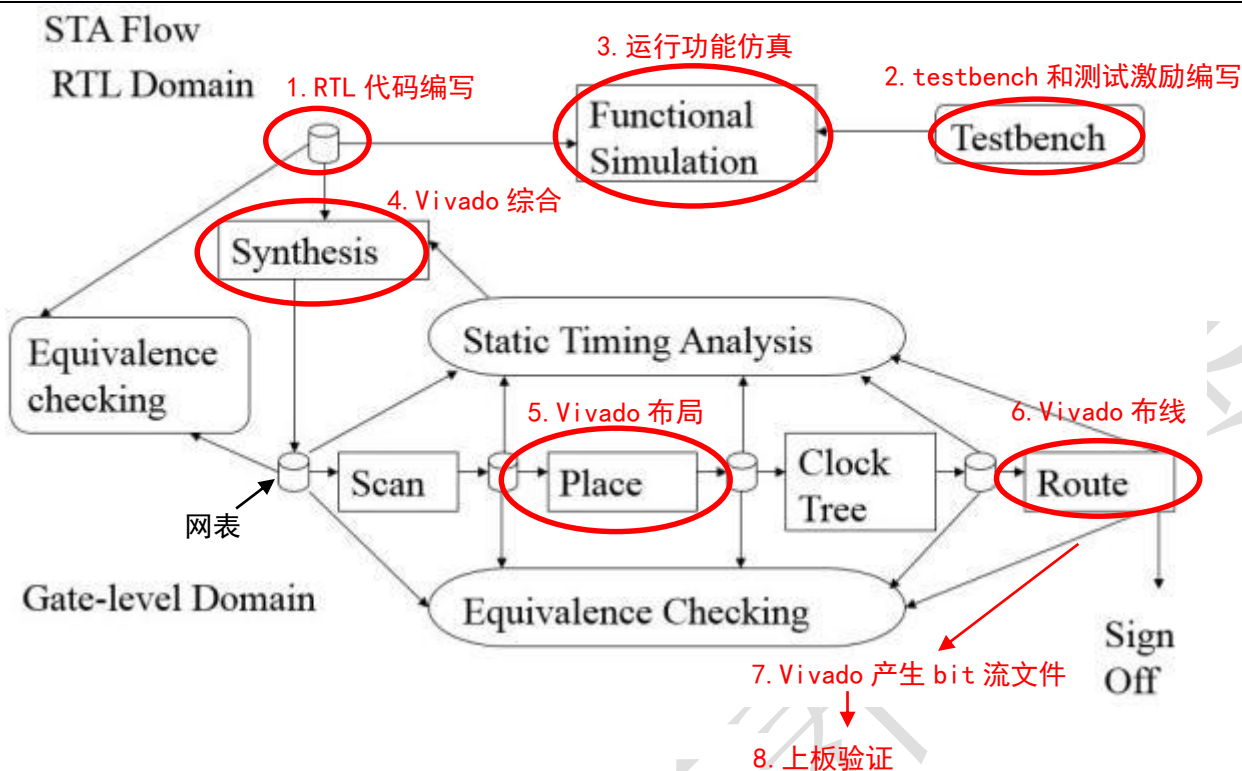


图 1-1 数字电路设计与 FPGA 开发对照图

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”。

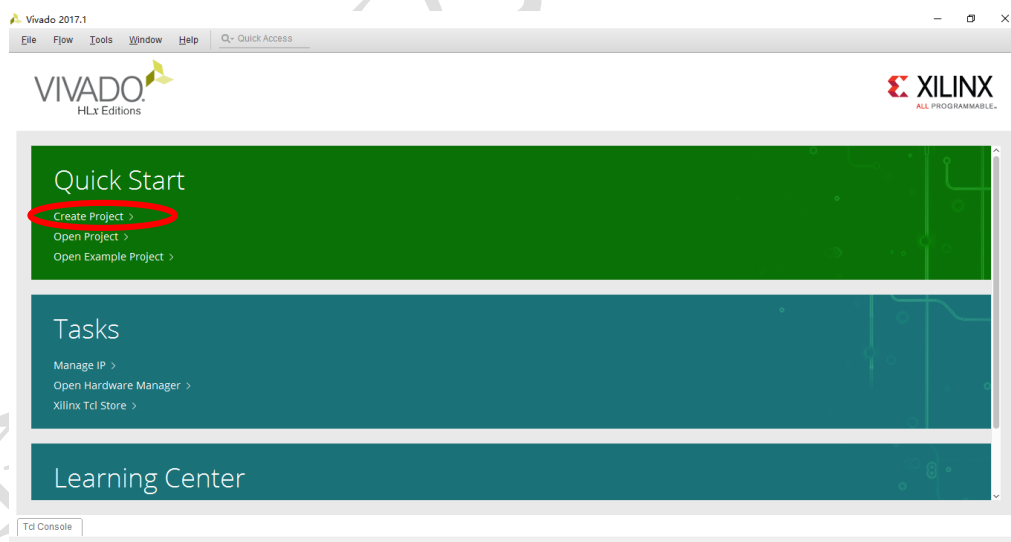


图 1-2 Vivado 启动界面

新建工程向导，点击“Next”。

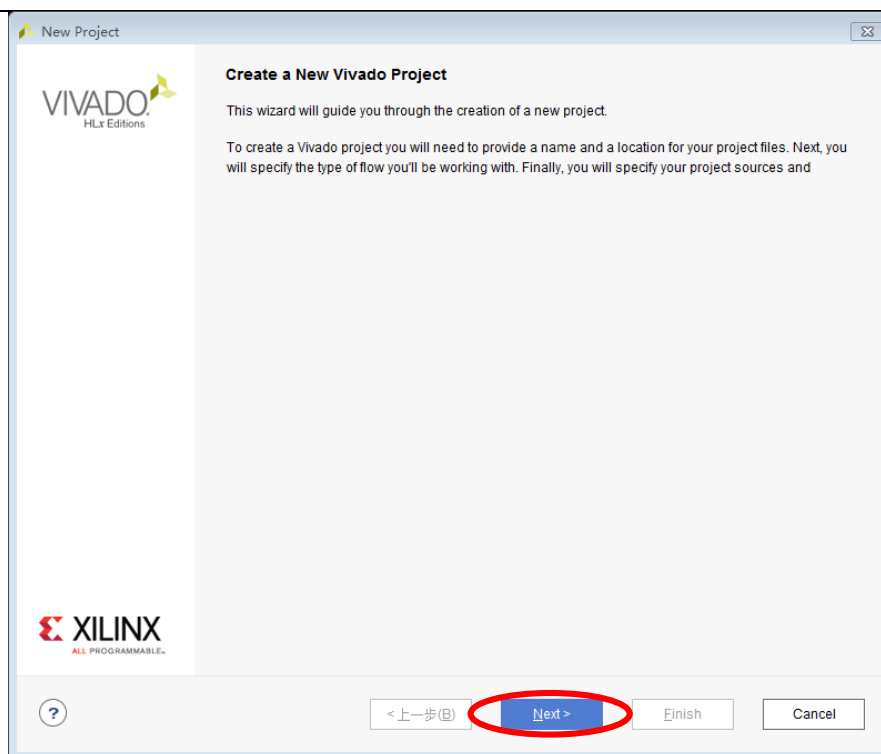


图 1-3 新建工程向导

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。

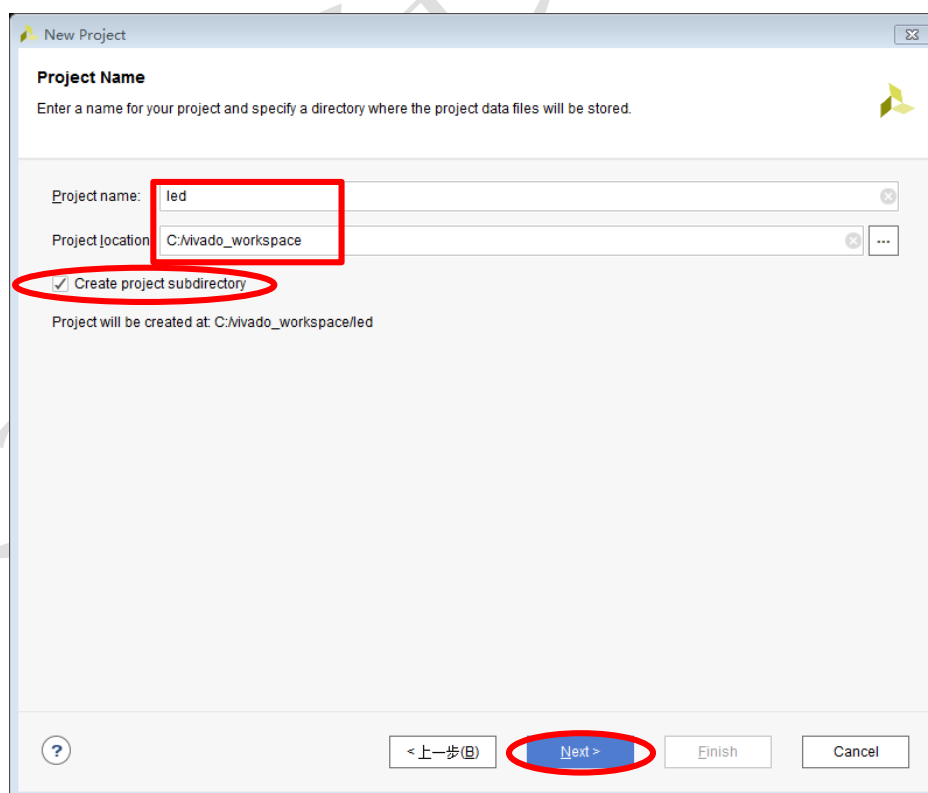


图 1-4 设置工程名称和位置

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”。

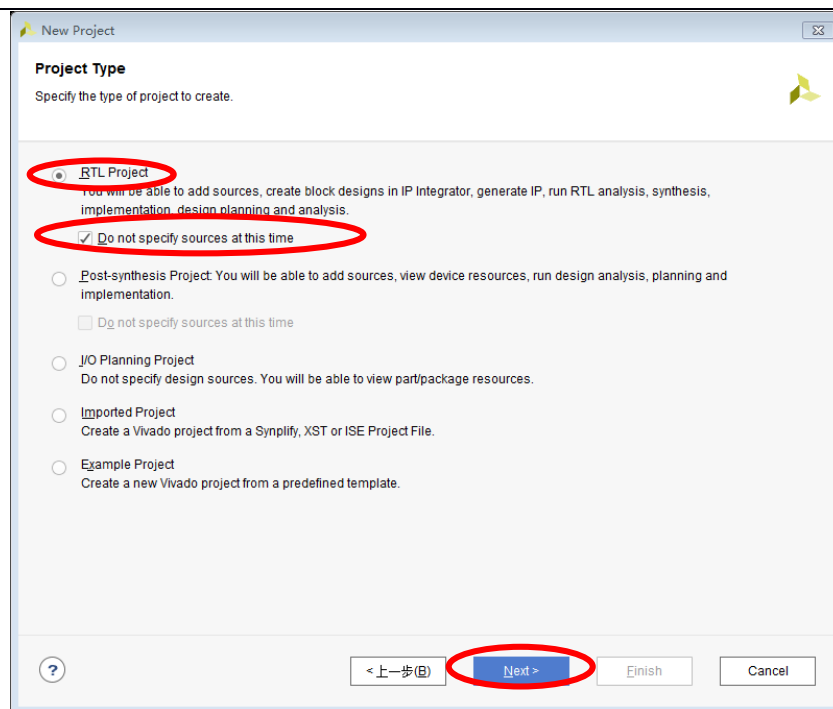


图 1-5 设置工程类型

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“xc7a200t**fbg676-2**”。点击“Next”。

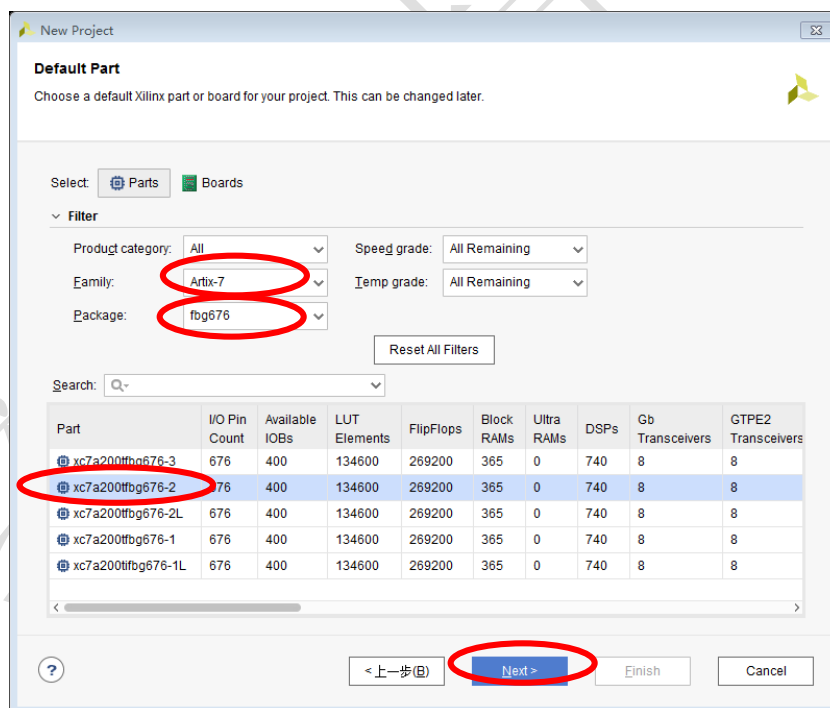


图 1-6 选择目标器件

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。

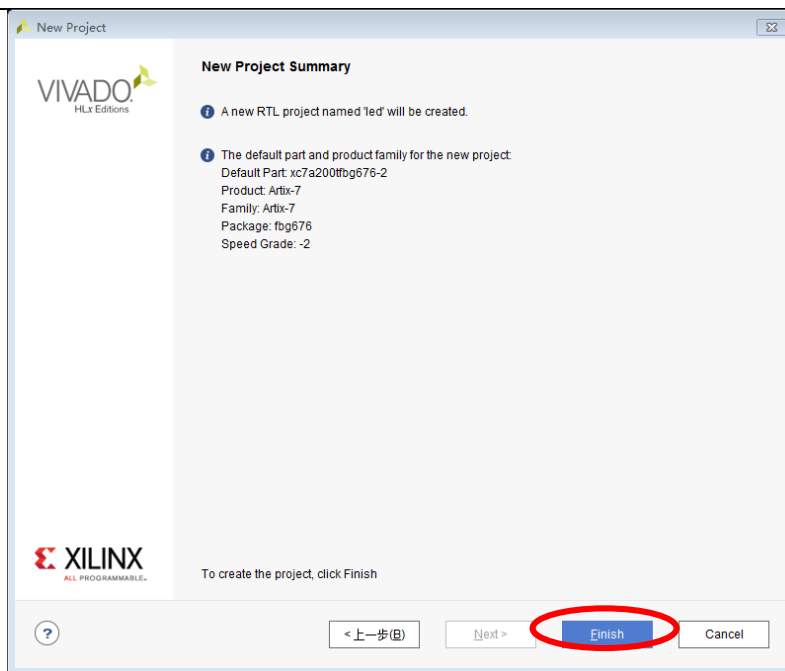


图 1-7 工程信息

完成工程新建。

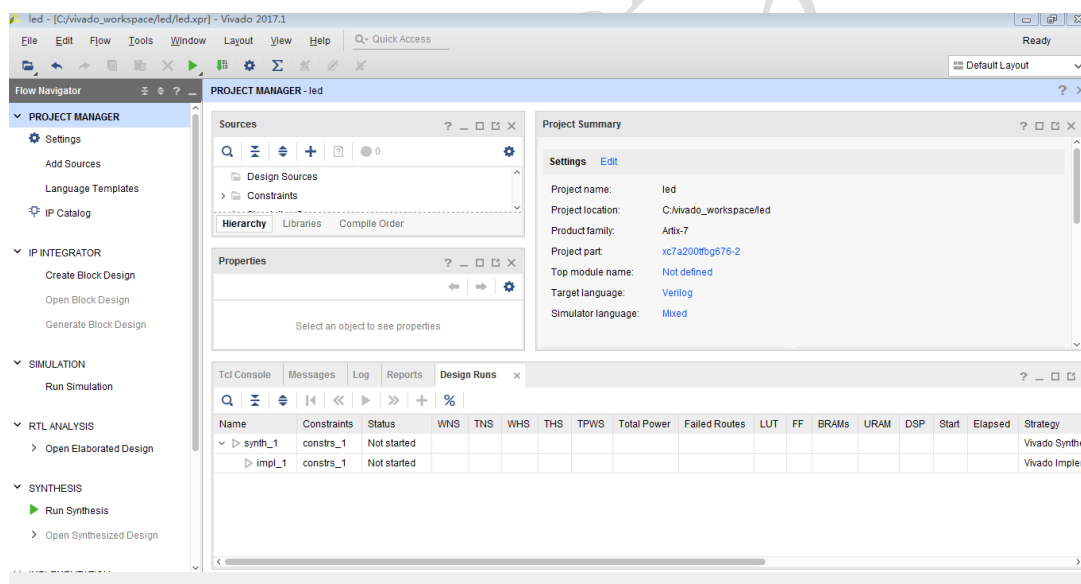


图 1-8 工程新建完成显示界面

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”。

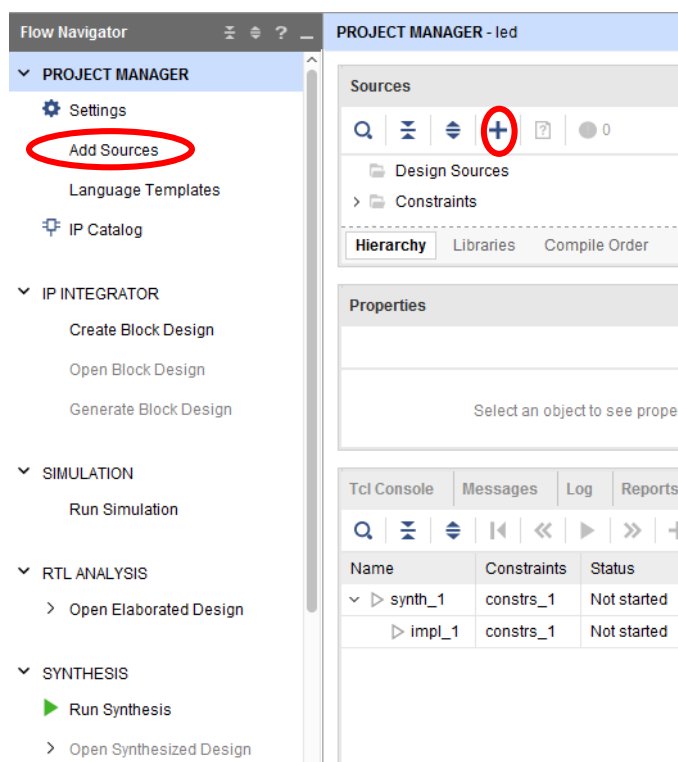


图 1-9 添加源文件

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”。

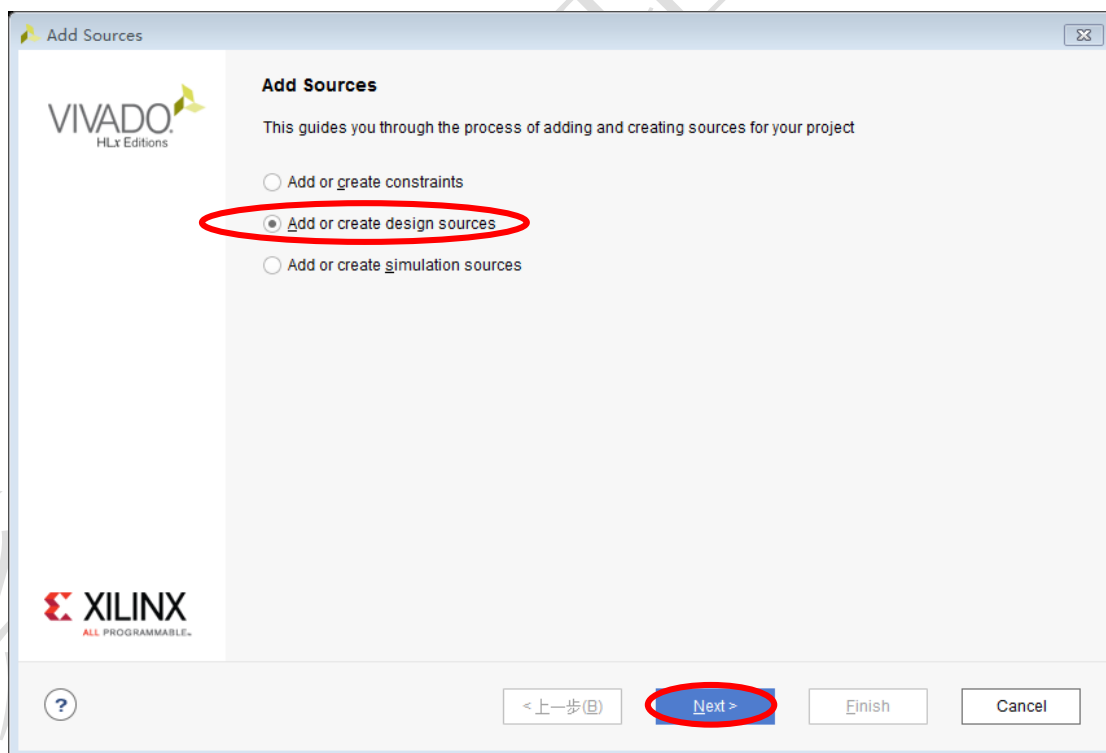


图 1-10 添加设计文件

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。

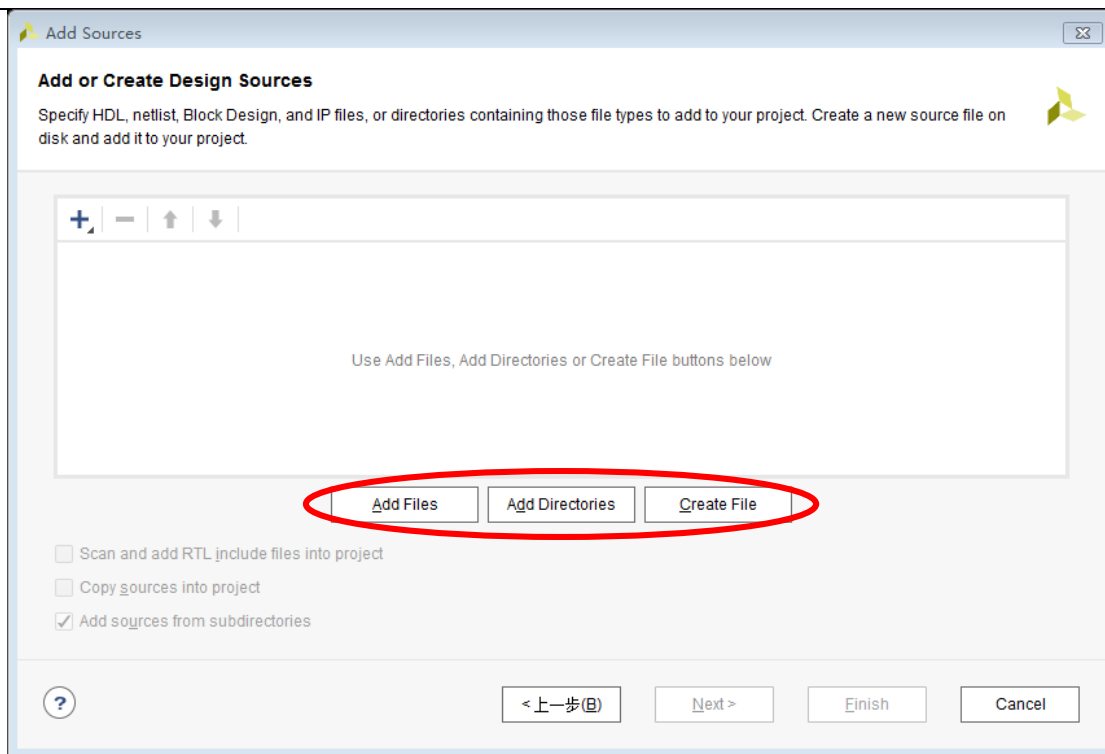


图 1-11 添加或者新建设计文件

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。

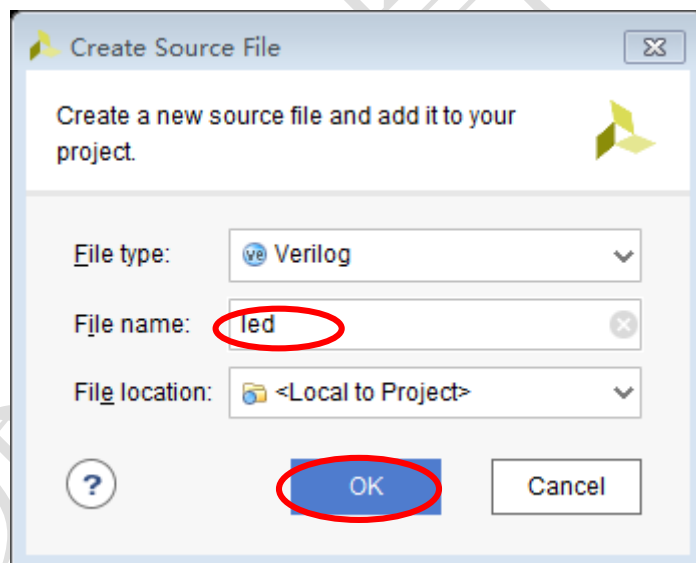


图 1-12 设置新的设计文件

继续添加设计文件或者修改已添加设计文件设置。点击“Finish”。

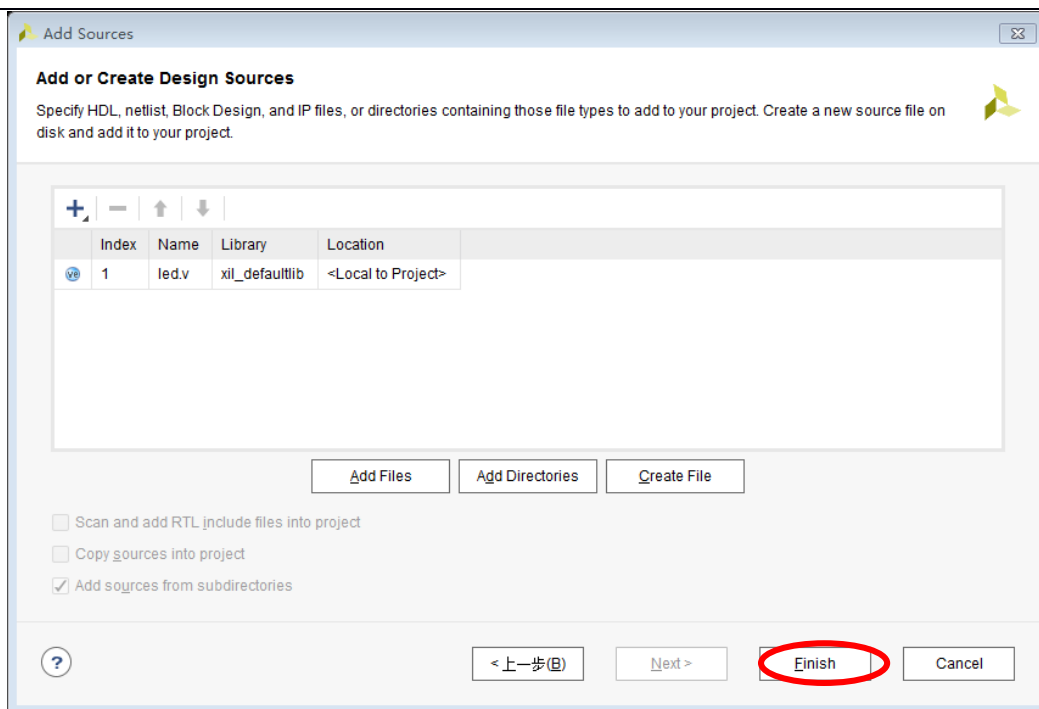


图 1-13 完成设计文件添加

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。

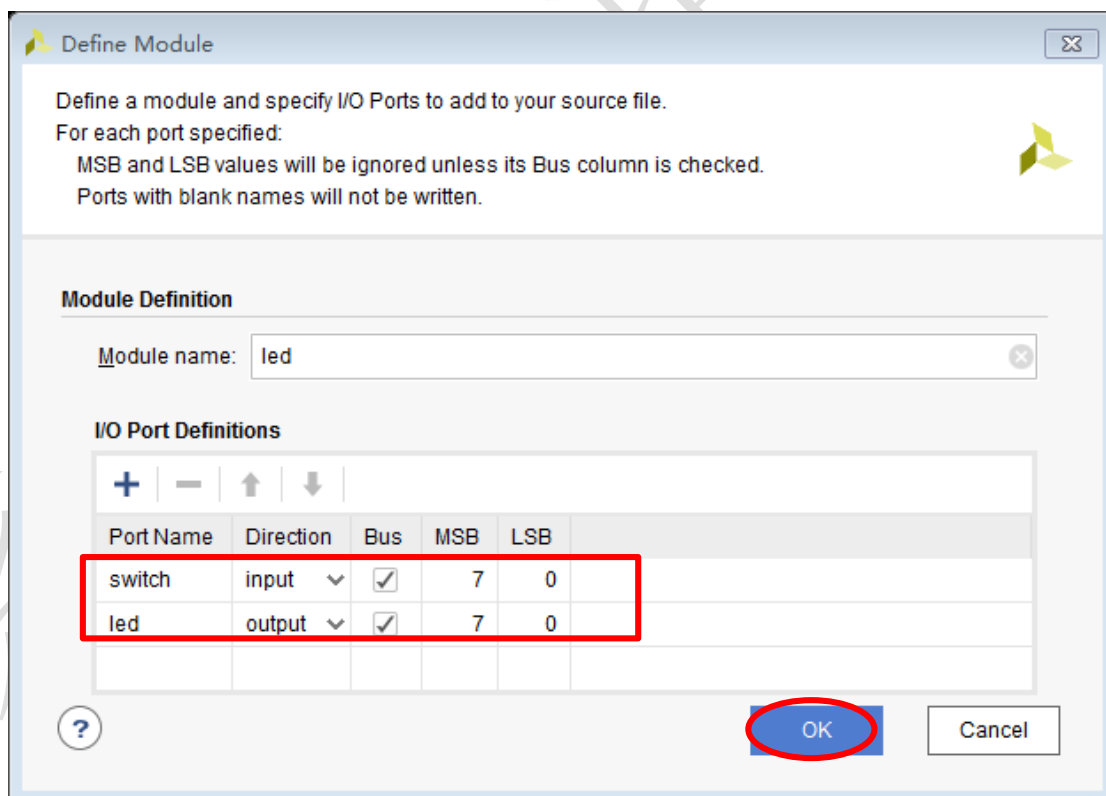


图 1-14 模块端口设置

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。

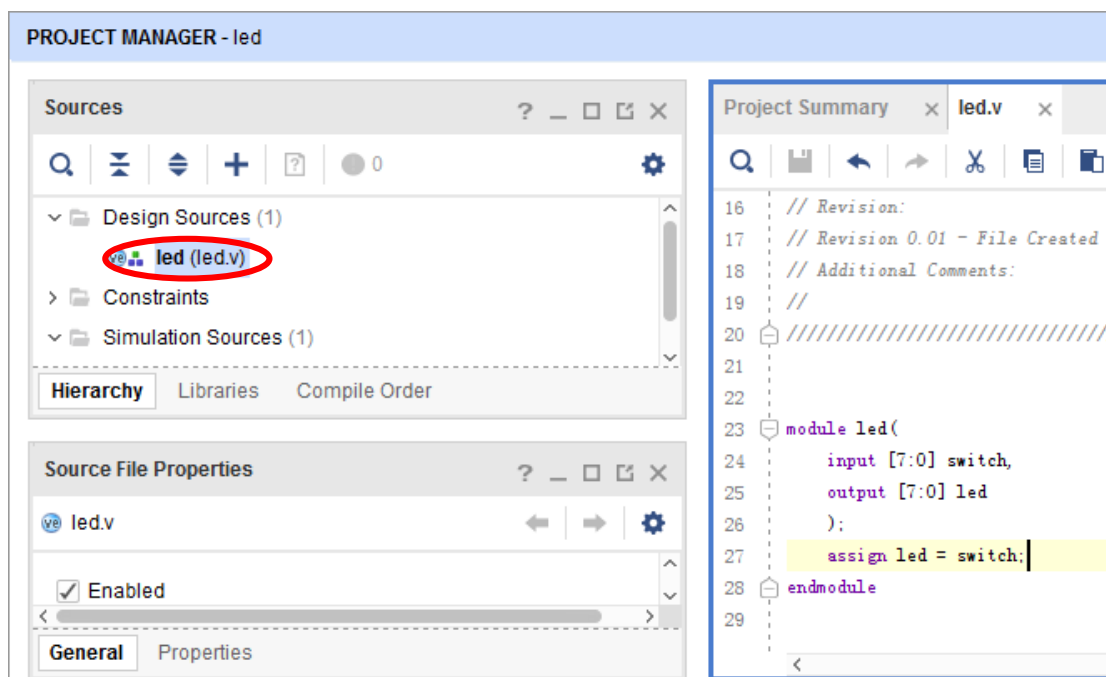


图 1-15 编辑设计文件

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”。

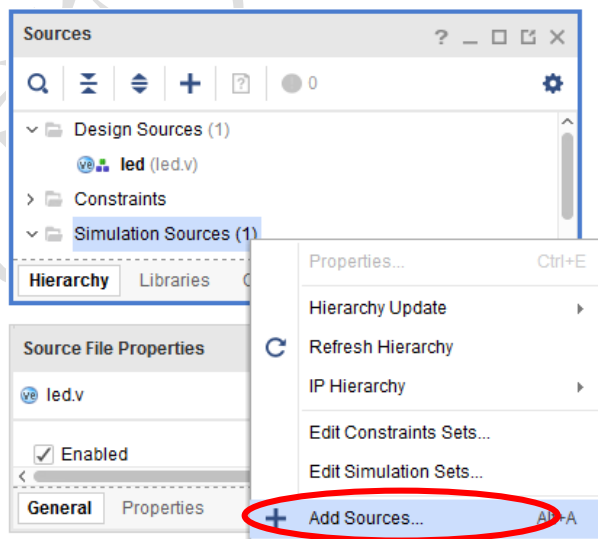


图 1-16 添加测试激励文件

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

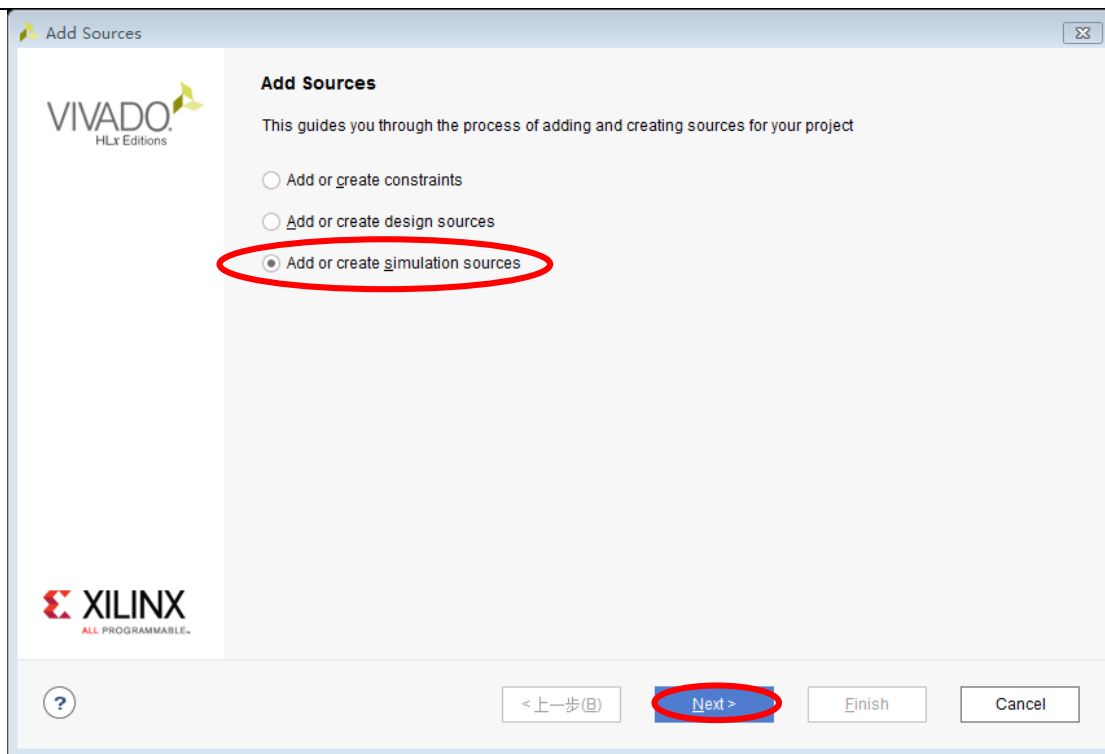


图 1-17 添加和新建测试文件

选择“Create File”，新建一个激励测试文件。

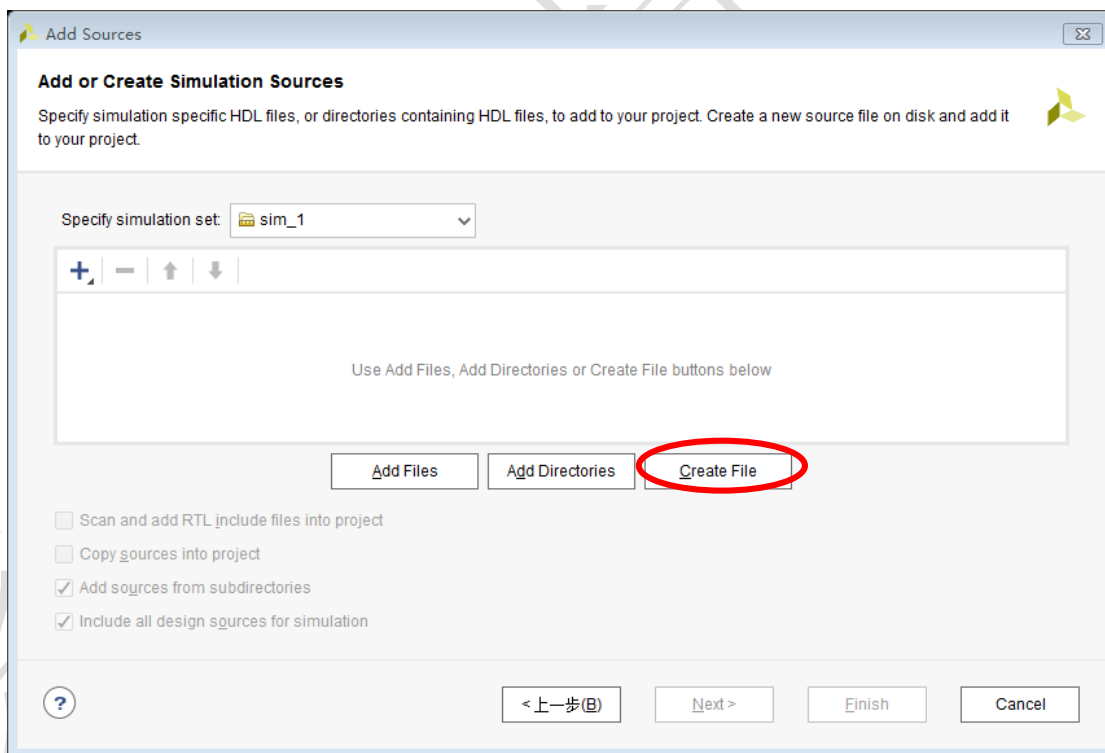


图 1-18 新建测试文件

输入激励测试文件名，点击“OK”。

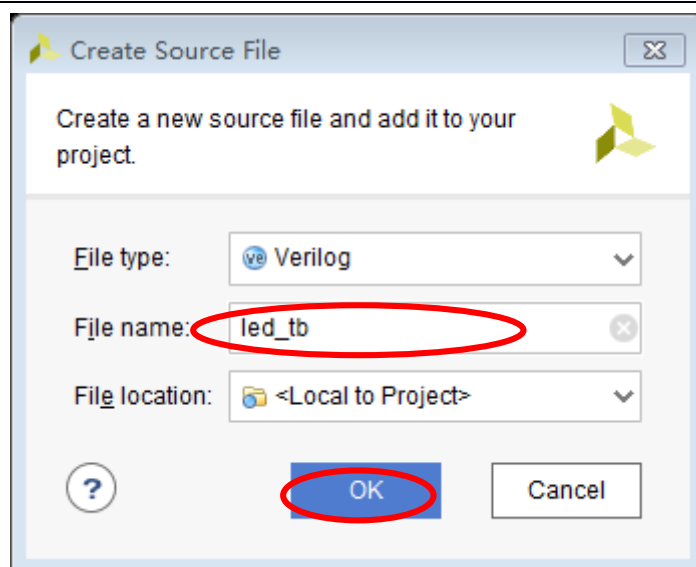


图 1-19 新建测试文件设置

完成新建测试文件，点击“Finish”。

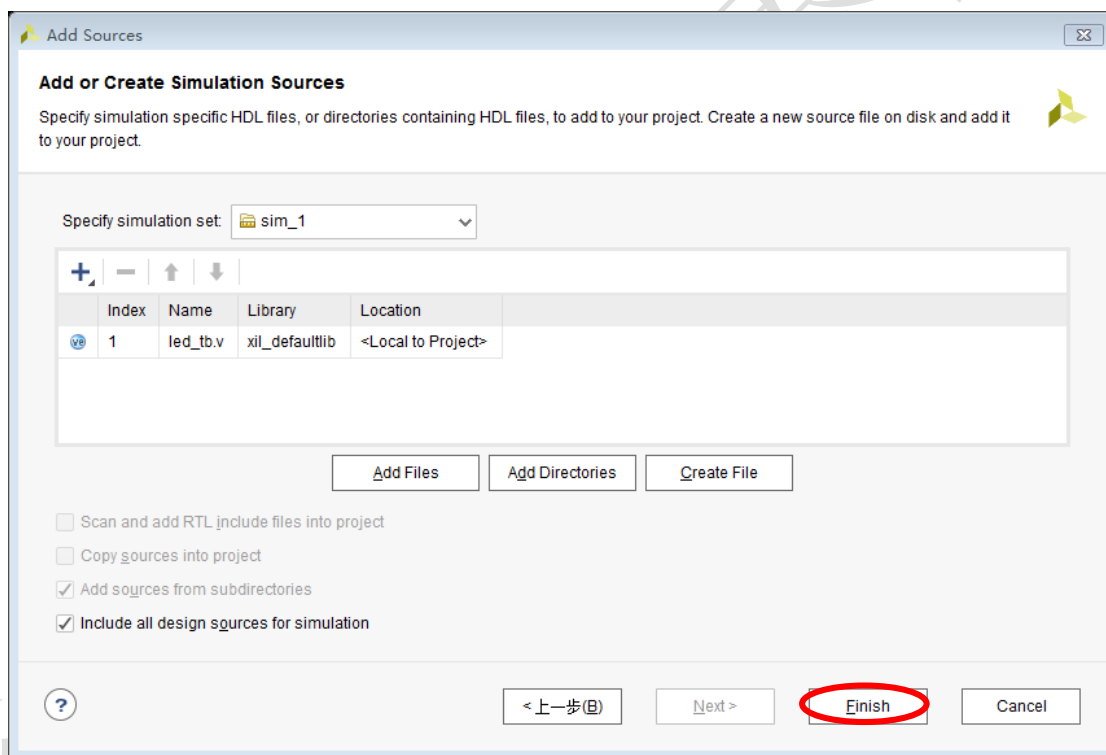


图 1-20 完成新建测试文件

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。

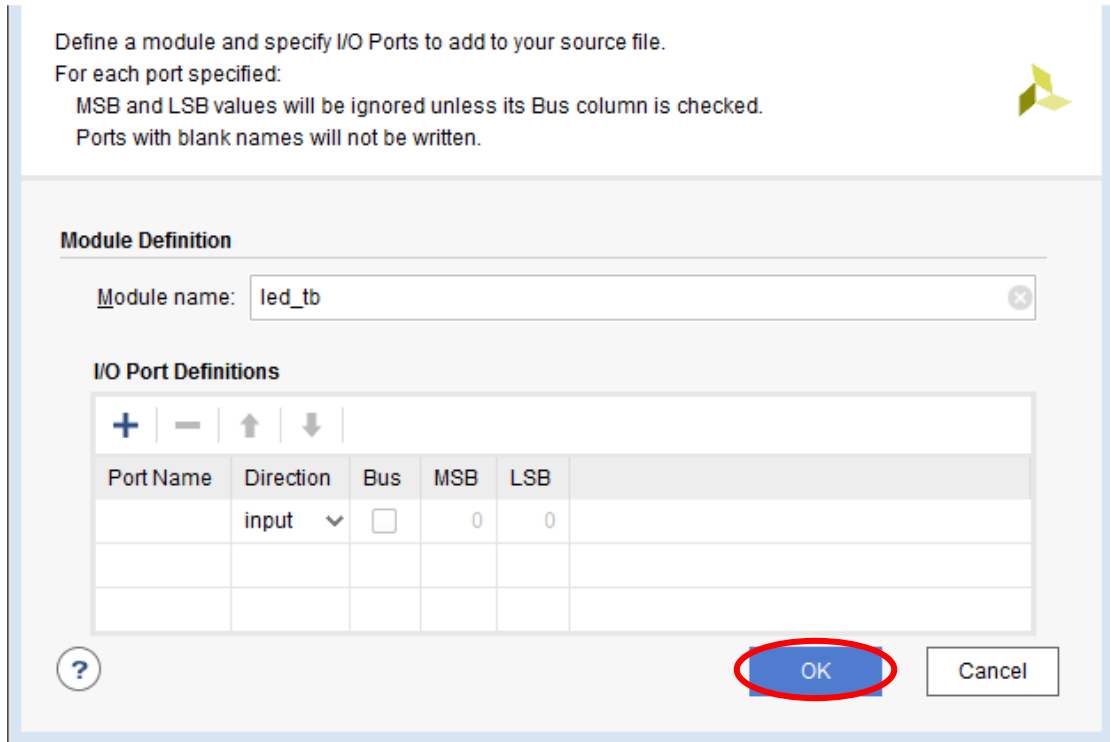


图 1-21 不需要进行端口设置

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。

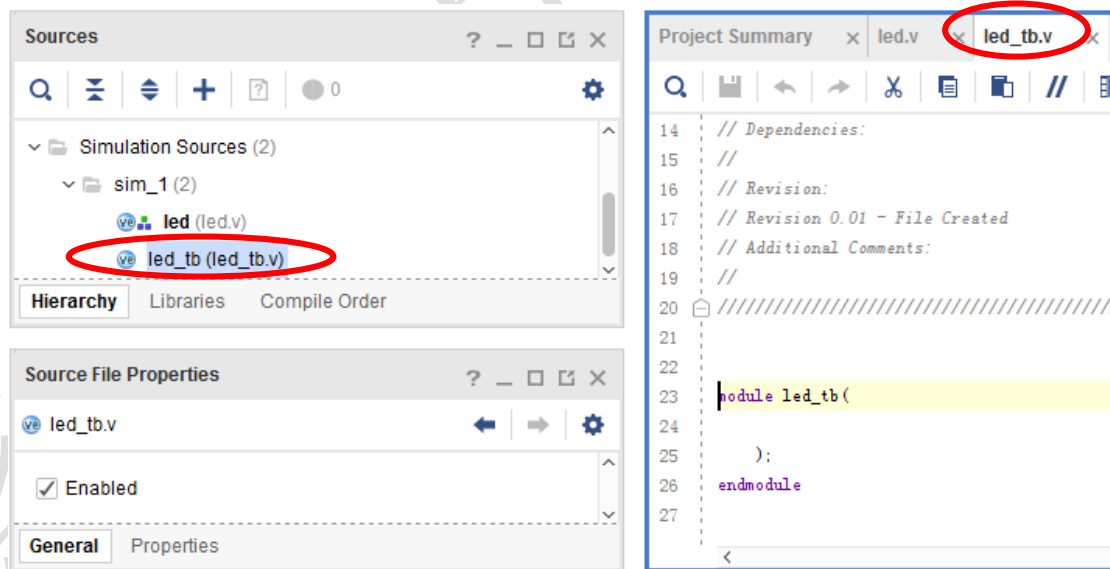


图 1-22 空白测试激励文件

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。

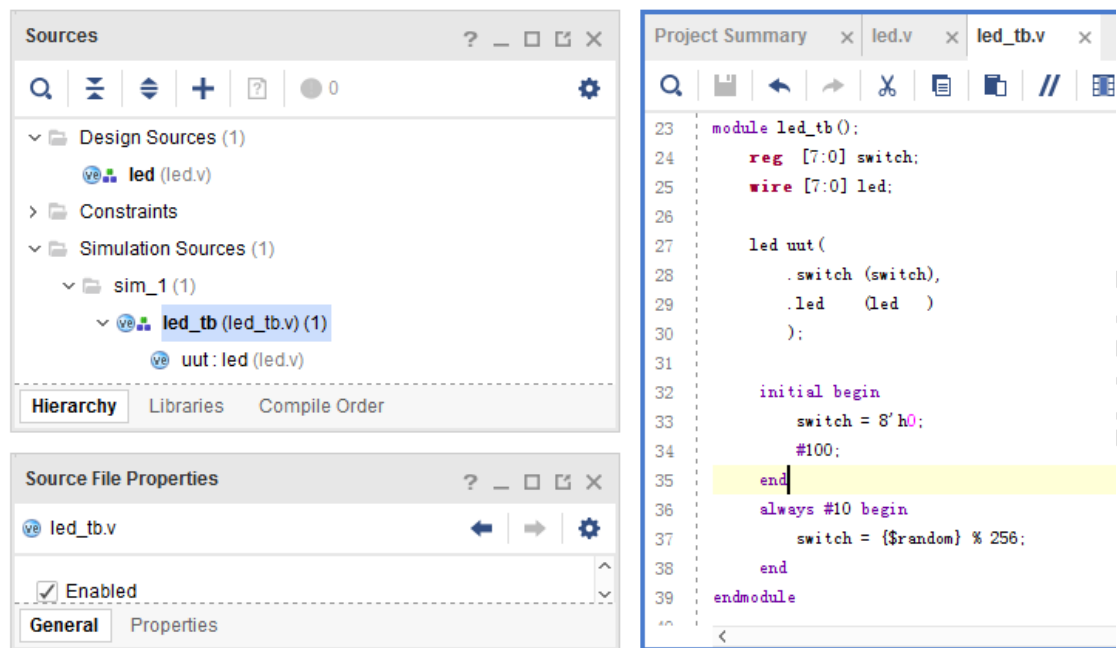


图 1-23 完成测试激励文件

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。

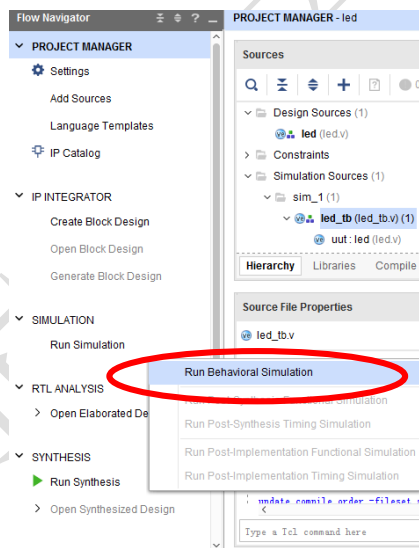


图 1-24 运行行为级仿真

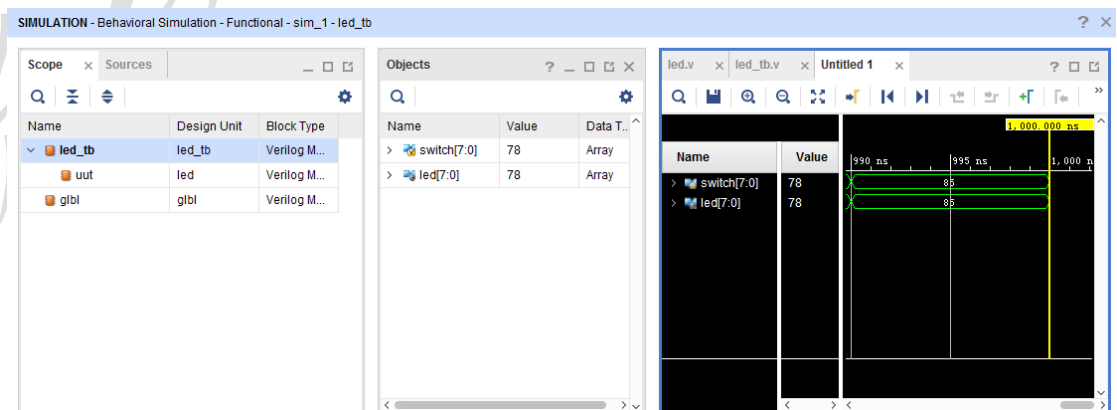


图 1-25 行为级仿真界面

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。

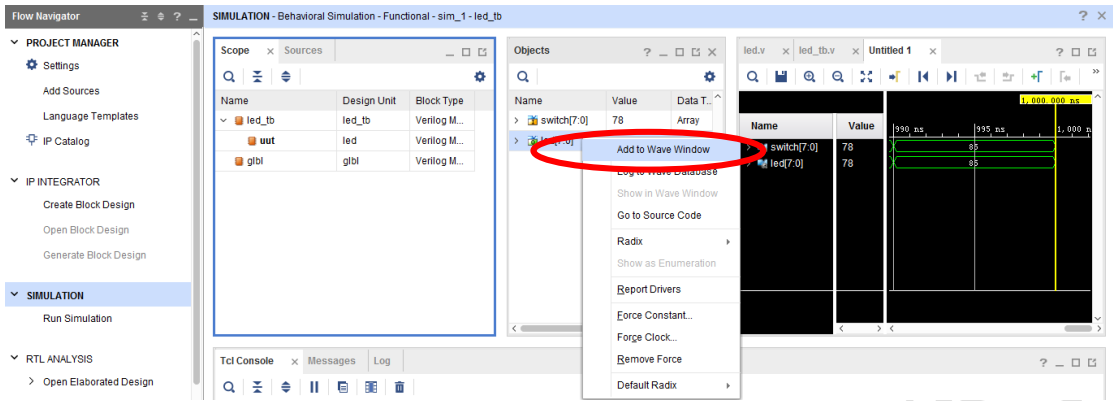


图 1-26 将内部信号添加到波形图

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。

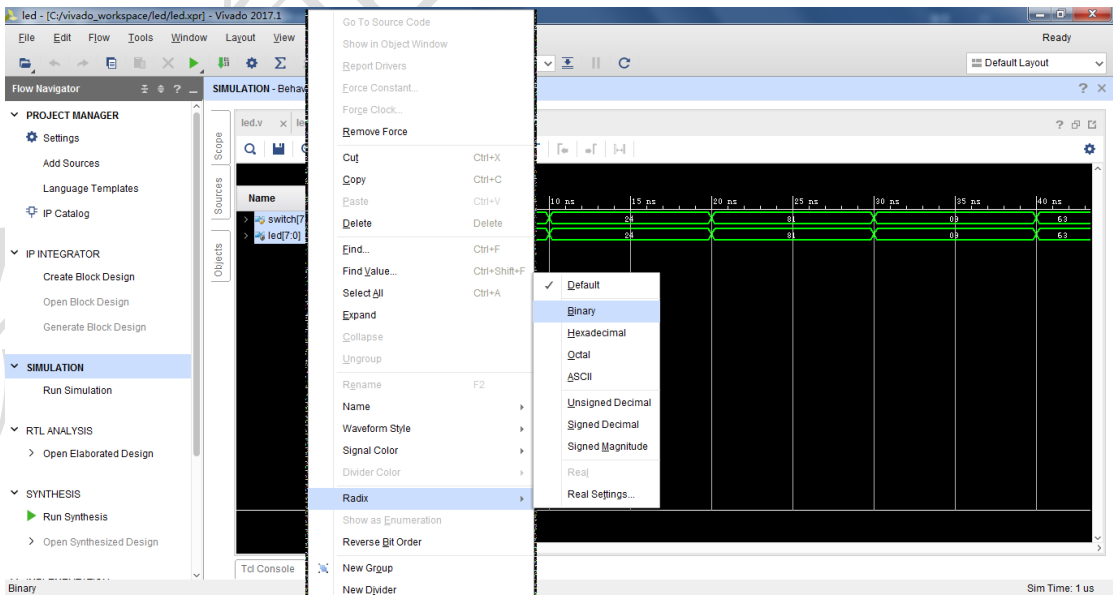


图 1-29 仿真波形窗口

查看波形检查设计功能正确性。

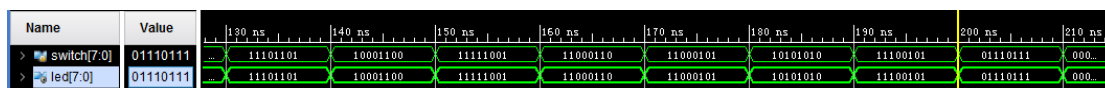


图 1-30 仿真波形结果

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。

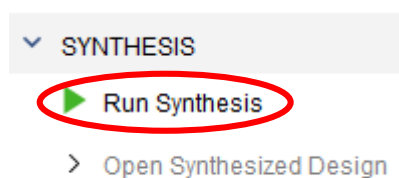


图 1-31 综合工程

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

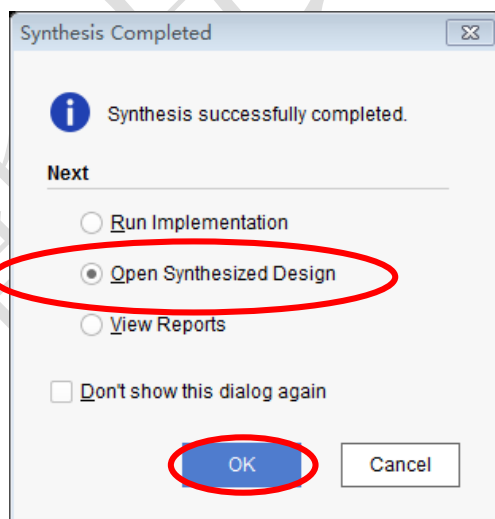


图 1-32 打开综合结果

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。