基于 HLS 的 FPGA 加速计算实验指导书

1.1. 实验目的

(1) 掌握基本高层次综合（High-Level Synthesis, HLS）工具使用方法；
(2) 初步了解 FPGA 逻辑电路的生成方式和计算方法；
(3) 学习 HLS 工具可综合的 C/C++编程和 FPGA 基本数据通路搭建；
(4) 通过完成基础的计算加速实验对 FPGA 和软硬件协同设计形成初步认识。

1.2. 实验设备

(1) 硬件：
PC 机、Zynq 7010 开发板。
(2) 软件：
Windows（或 Ubuntu）操作系统；
Xilinx 公司 Vivado、Vitis HLS（或 Vivado HLS）集成开发环境。

1.3. 实验准备

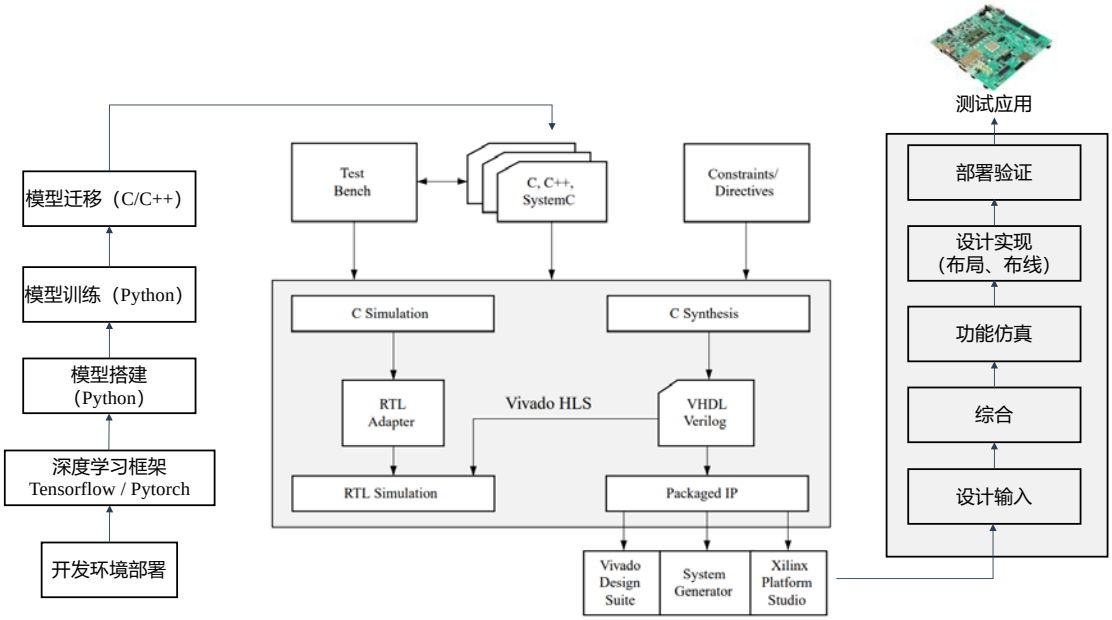
(1) 在 PC 上正确安装 Vivado（或 Vitis）软件；
(2) 掌握 C/C++、Verilog HDL 基本语法，具备初级编程能力。

1.4. 实验原理

近年来，随着半导体制备工艺逐渐逼近物理极限，CPU 等通用处理器的性能提升速度日趋缓慢，为满足大数据时代下，各行各业对大规模数据的高能效计算需求，以 FPGA 为代表的可重构计算体系成为众多新兴领域的关注热点。尤其是在以卷积神经网络应用为基础的边缘智能计算领域，FPGA 作为一种备受关注的新型计算形态，在实时、高能效，且需要灵活应对各种应用场景的边缘端设备中有着极其重要的技术价值和广泛的应用前景。

本实验以典型的矩阵运算为基础，演示基于 HLS 的 FPGA 计算部署和应用方法，旨在帮助学习者理解 FPGA 加速计算设计，熟悉 Xilinx 工具链和设计流程。

基于 HLS 的 FPGA 加速计算设计和部署流程如下图所示。



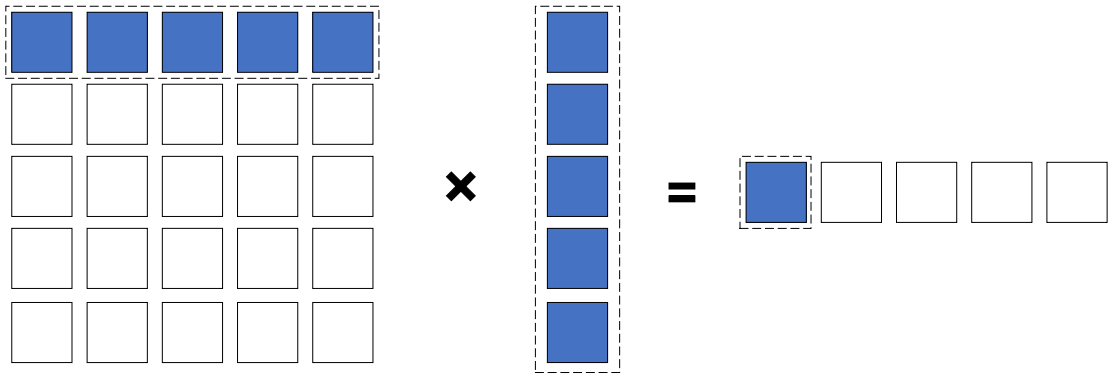
高层次综合工具可以在一定程度上屏蔽复杂的底层硬件设计，使设计者更加高效的完成硬件架构设计。

首先，使用 C/C++语言描述计算原理，其语法和软件编程略有不同。目的是描述硬件计算过程。

其次，使用 HLS 工具对 C/C++设计进行高层次综合，实现 RTL IP 核逻辑。为验证逻辑设计的正确性，通常会在综合前进行 C 功能仿真或 RTL 与 C 的协同仿真。

最终，生成可部署于偏上系统的 IP 核，通过 Vivado 硬件设计工具，进行 FPGA 片上系统设计，连接所需的数据接口和控制模块，构建数据通路，部署在芯片上进行实际的功能验证。

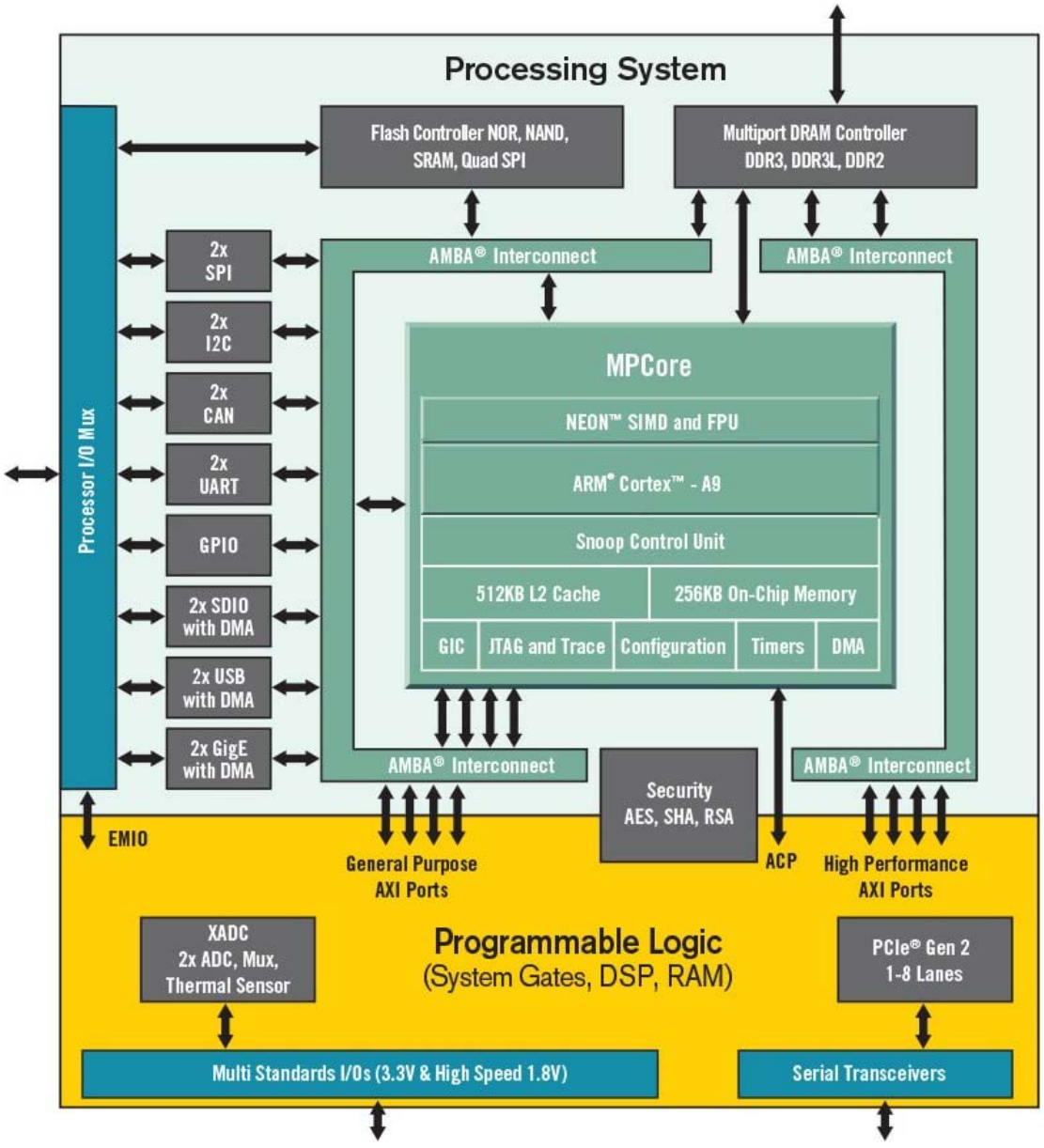
本次实验从 C/C++设计环节开始，进行矩阵乘法的 FPGA 加速计算设计。矩阵运算原理如下图所示。



实验采用尺寸为 5×5 的矩阵数据 A 作为输入，与向量 b 进行逐一元素的乘累加计算，最终得到结果 c。该过程虽然简单，但其是目前深度学习领域卷积神

经网络最基本的运算结构，是计算密集型任务的典型代表。

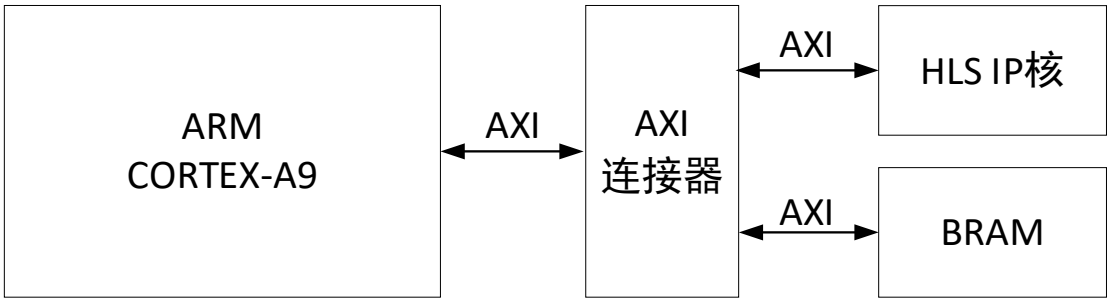
在矩阵运算设计的基础上，本次实验结合 Zynq 处理器进行实际的上板实验。Zynq 是 Xilinx 公司提出的一款可编程的 SoC 处理器，采用 ARM+FPGA 的异构计算架构，兼具灵活性和通用性，是目前实现定制化嵌入式可重构计算应用的主流平台之一。Zynq 的基本架构如下图所示。



Zynq 由 PS 和 PL 两部分构成，PS 主要有 ARM 通用处理器构成，在本次实验中负责顶层的数据调度；PL 主要由可编程逻辑块构成，是本次实验 IP 核硬件实现的主要平台。

Zynq 系统中数据的传输主要依赖于 AXI 总线，芯片内集成了丰富的布线资源，可支撑灵活、高效的多通道数据互传。此外，芯片内部可由 BRAM 构建片上存储单元，实现数据的片上存储，减少片外互联总线的传输压力。

本实验基于 AXI_Master 总线构建数据通路，数据处理流程如下图所示。

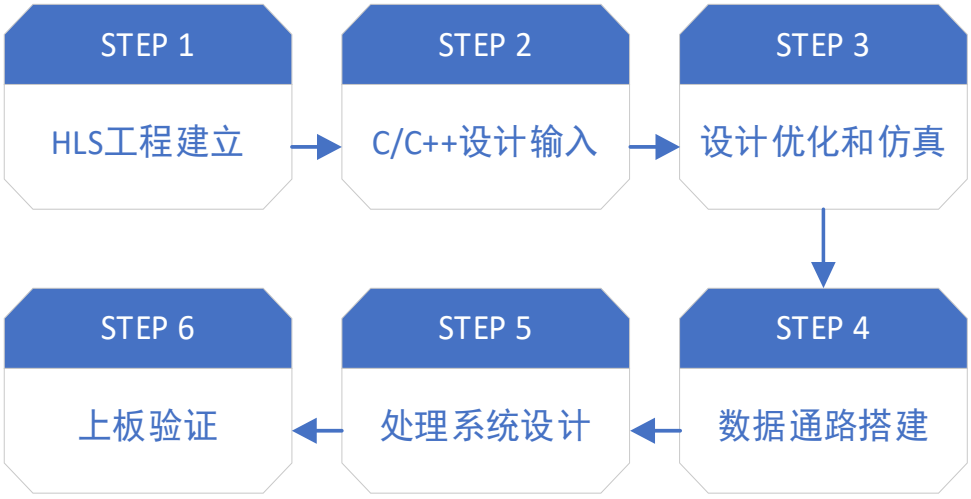


ARM 通用处理器内核首先对外设进行初始化配置，并将输入参数 A 写入 BRAM 中，然后由 HLS IP 核从 BRAM 中自动读取参数并进行计算，产生结果后再通过 AXI 总线写回 BRAM 中，从而完成一次处理过程。

在片上资源和数据传输速度允许的情况下，计算系统可实现更大数据量的连续计算，从而完成类似于卷积神经网络的高密度计算负载型任务。

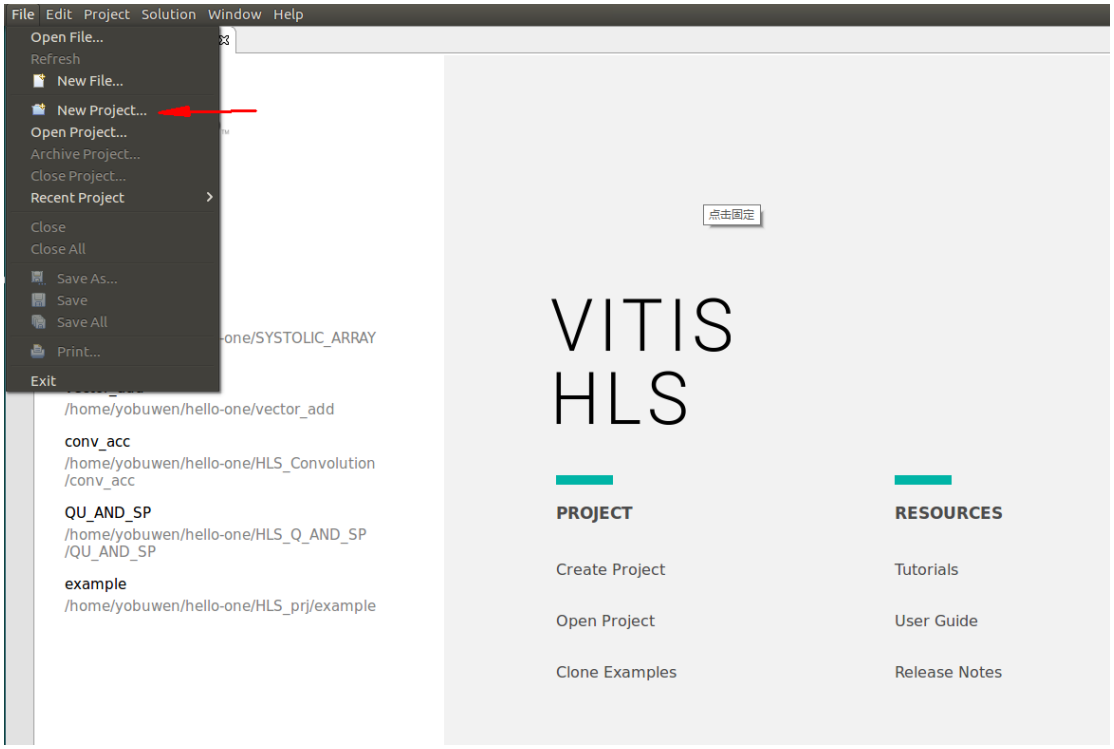
1.5. 实验内容

实验流程如下图所示。

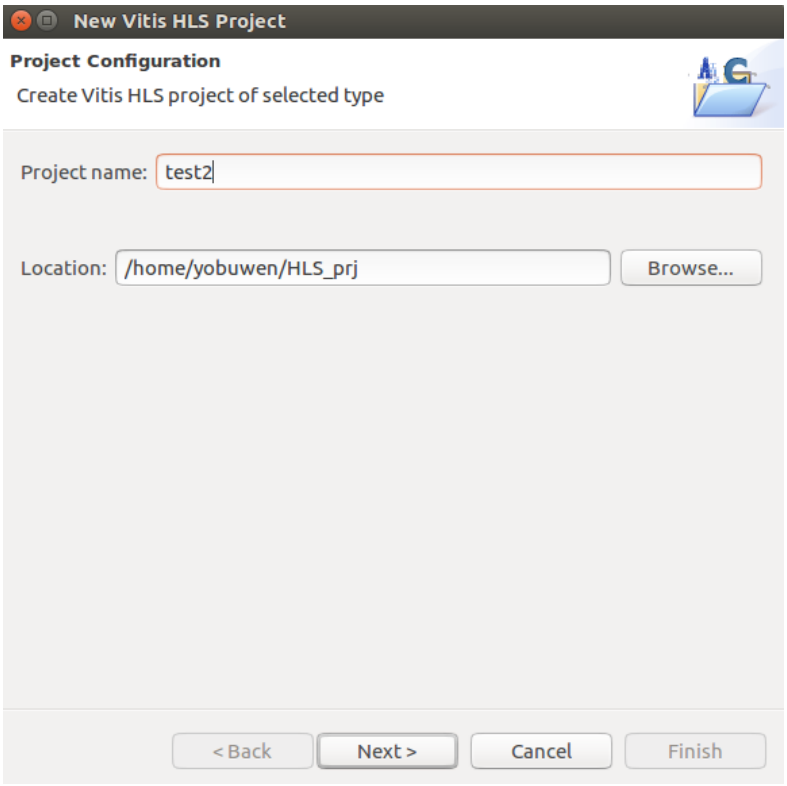


STEP1. HLS 工程建立

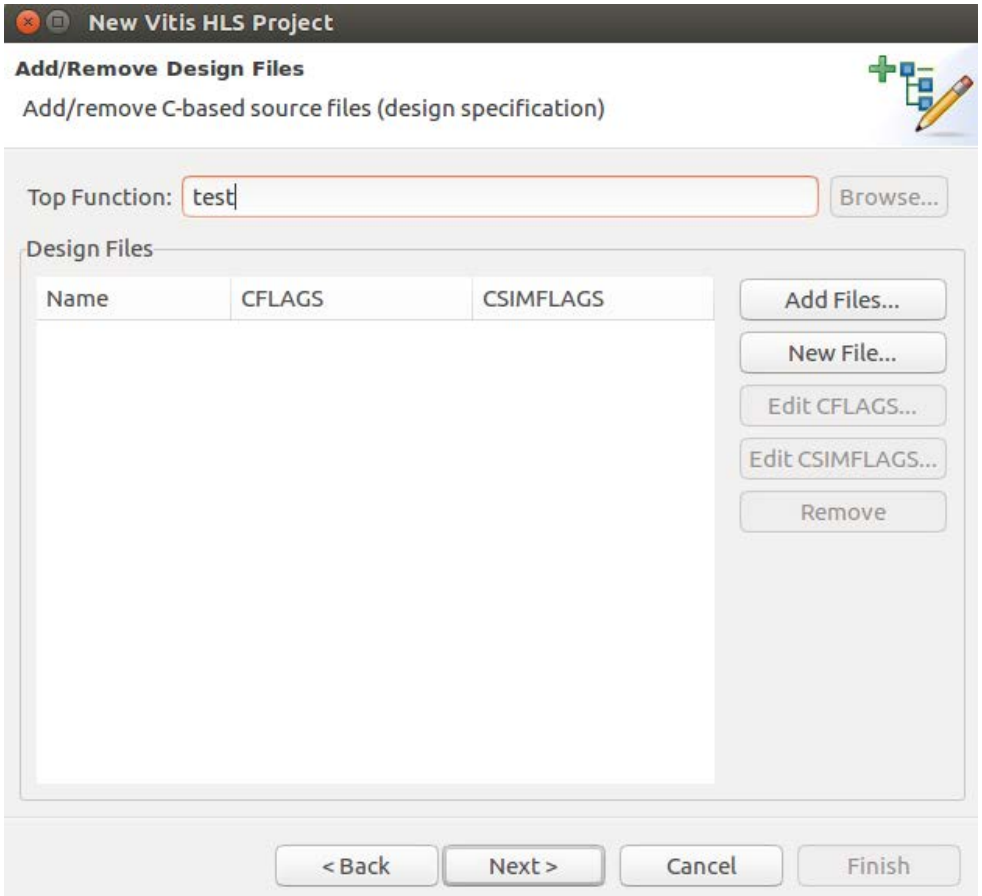
打开 Vitis HLS，新建一个工程，如下图所示。



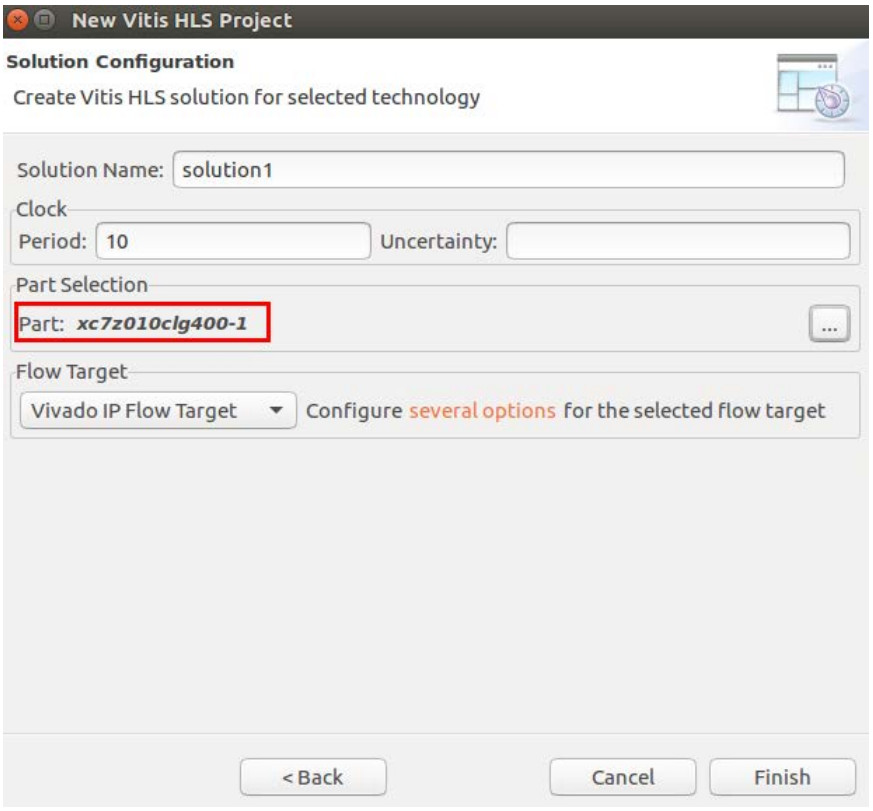
填写工程名，并选择工程存储路径，点击 NEXT。



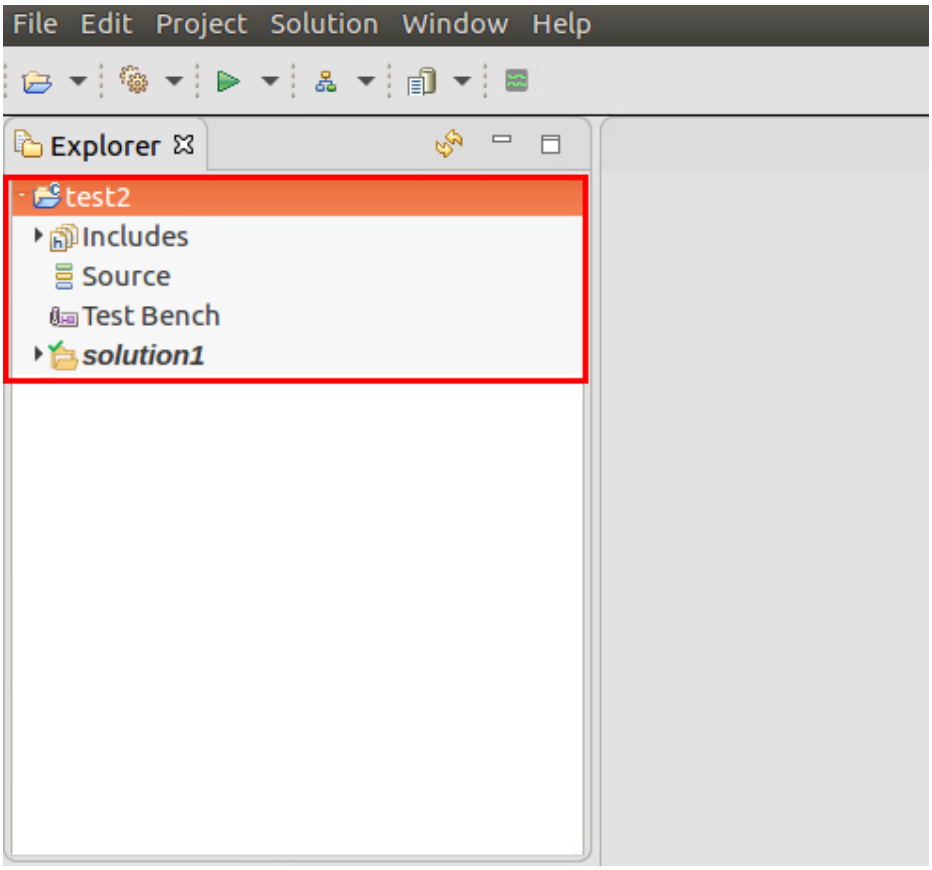
填写要综合的硬件模块的顶层函数的函数名，点击 NEXT。



Testbench file 可以后续添加，此处 NEXT 跳过，设置 solution name 并选择目标硬件，点击 FINISH。

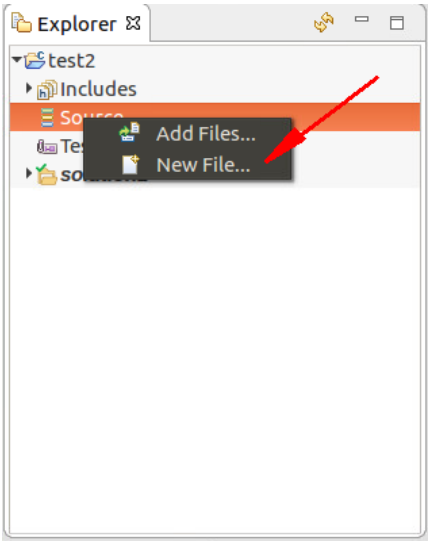

New Vitis HLS Project
Solution Configuration
Create Vitis HLS solution for selected technology
Solution Name: solution1
Clock
Period: 10
Uncertainty:
Part Selection
Part: xc7z010clg400-1
...
Flow Target
Vivado IP Flow Target
Configure several options for the selected flow target
< Back
Cancel
Finish

HLS 工程建立完成。


File Edit Project Solution Window Help
Explorer
test2
Includes
Source
Test Bench
solution1

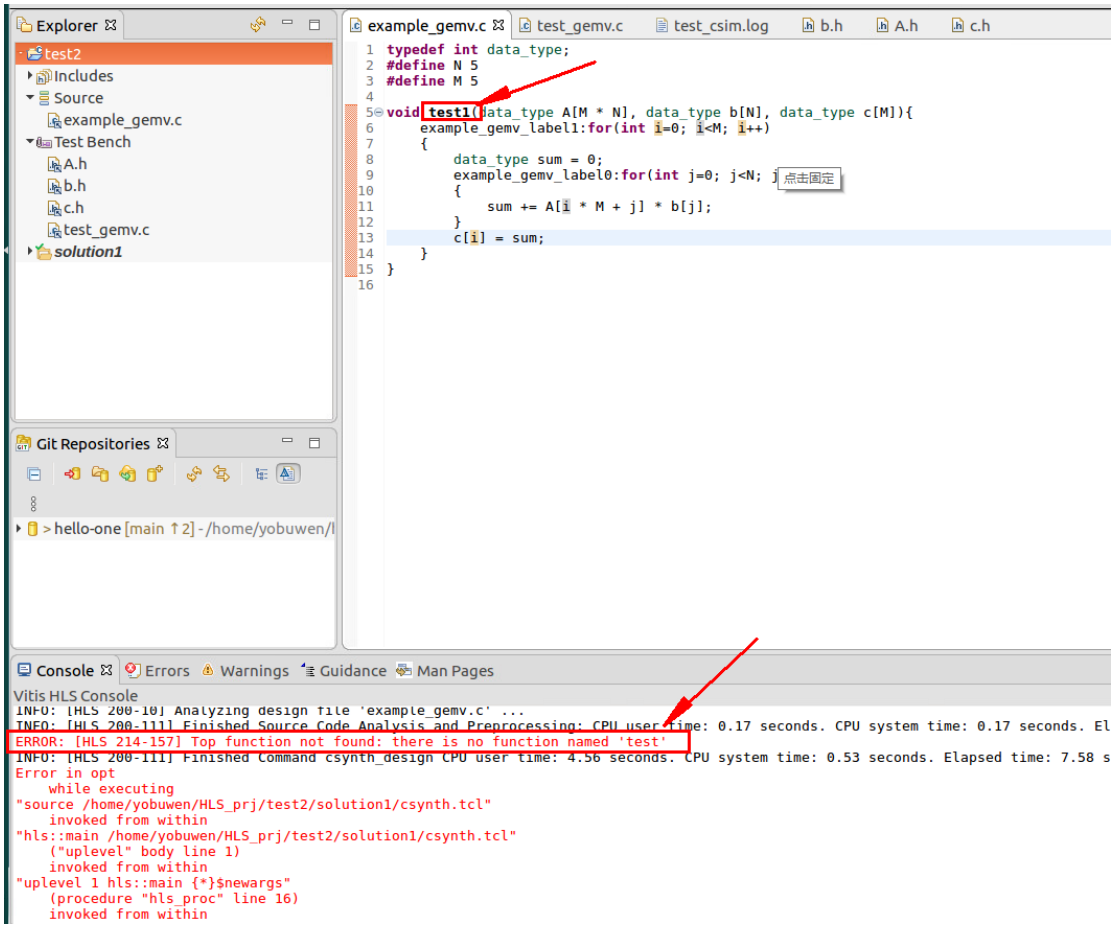
STEP2. C/C++设计输入

在 Explorer 窗口中，右键 Source 选择 New Files，输入 C 语言设计源码（也可以直接添加文件）。

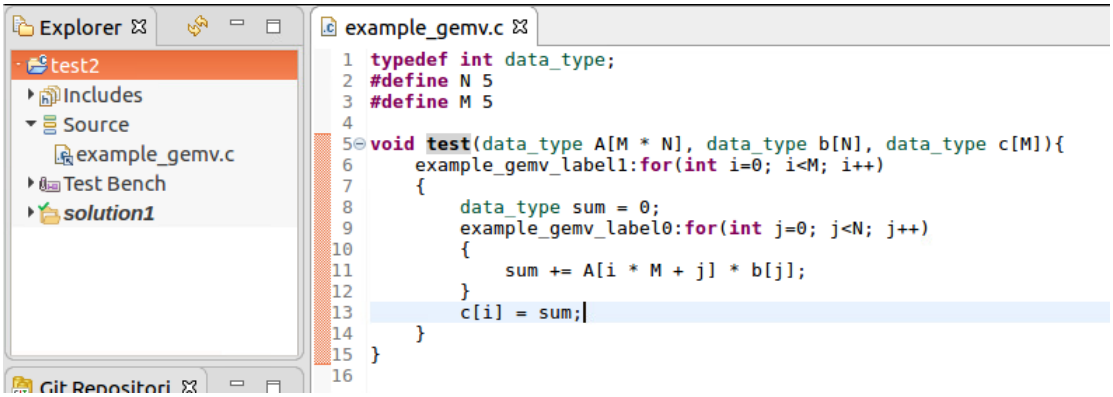


注意：输入文件顶层函数名，必须和 Top function 一致，否则在综合是编译器会报错。

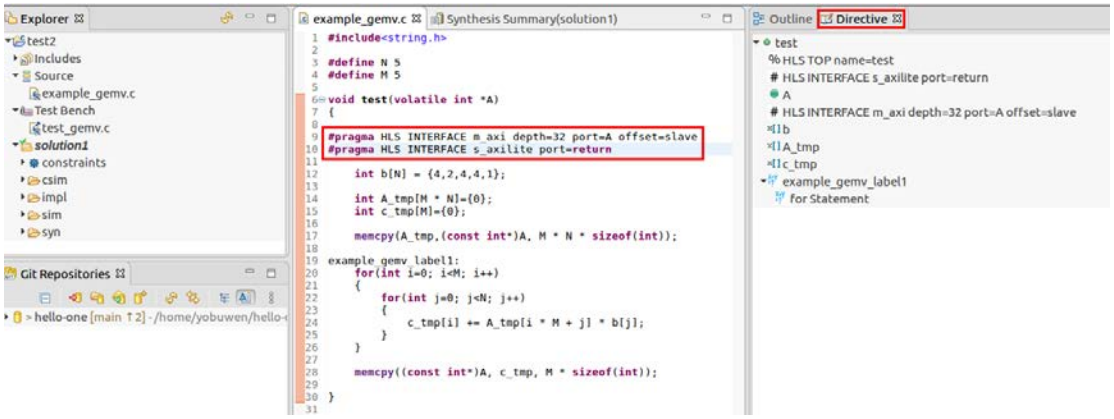
函数名不一致报错如下图所示：



正确输入，如下图所示：

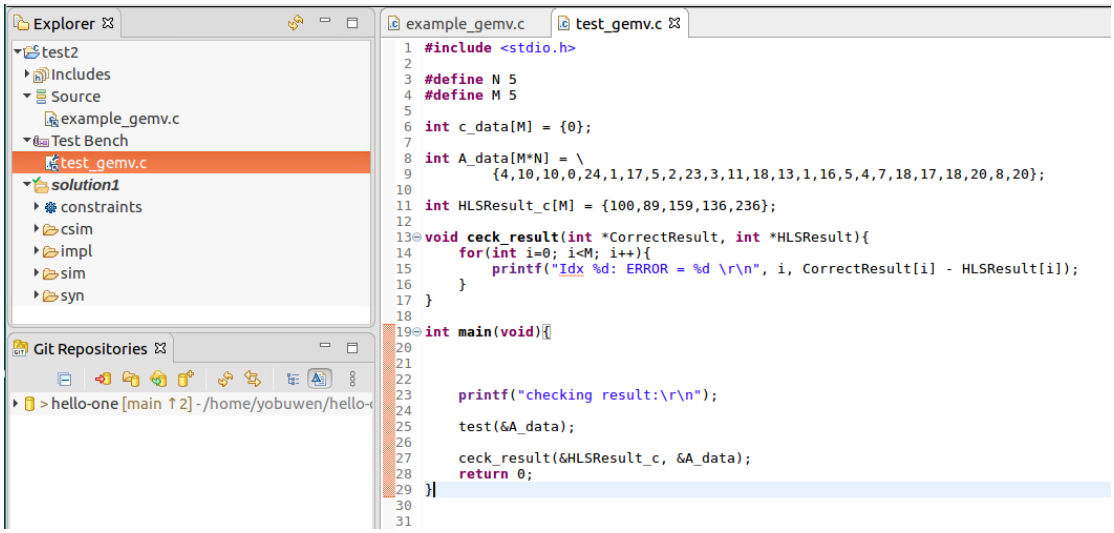


HLS 最终目标是形成 FPGA 的底层逻辑，实现硬件计算功能，因此在设计中添加总线接口，用来实现数据的输入和输出，如下图所示。



本设计采用 AXI 总线接口，实现参数的输入和计算结果的读取。#pragma 命令可以通过手动输入，也可通过右侧 Directive 菜单输入。

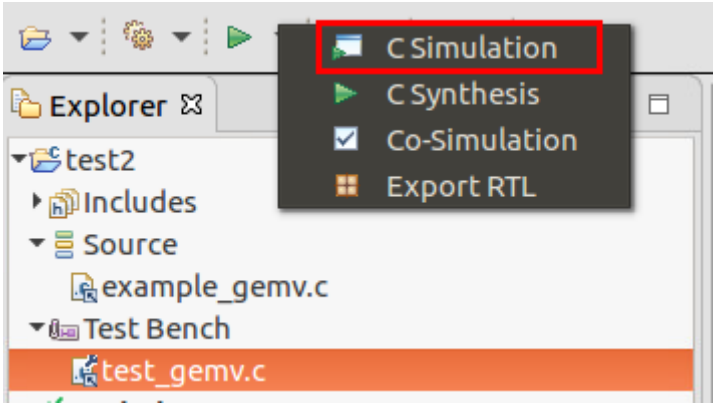
在 Explorer 窗口中，右键 Test Bench 选择 New Files，输入 C 语言测试代码（也可以直接添加文件）。



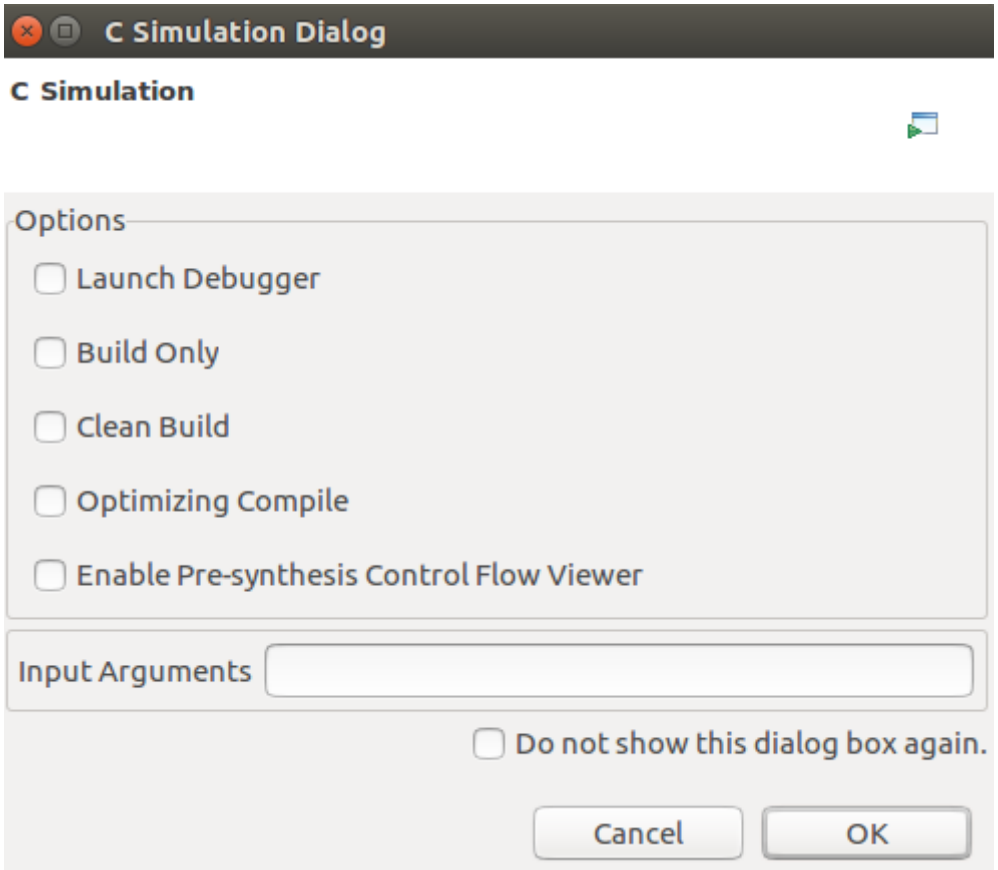
至此，HLS 设计输入完成。

STEP3. 设计优化和仿真

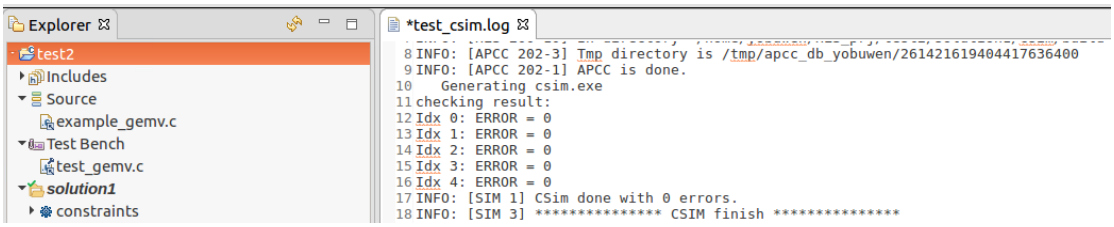
完成设计输入后，首先进行 C Simulation，验证 C 设计正确性。



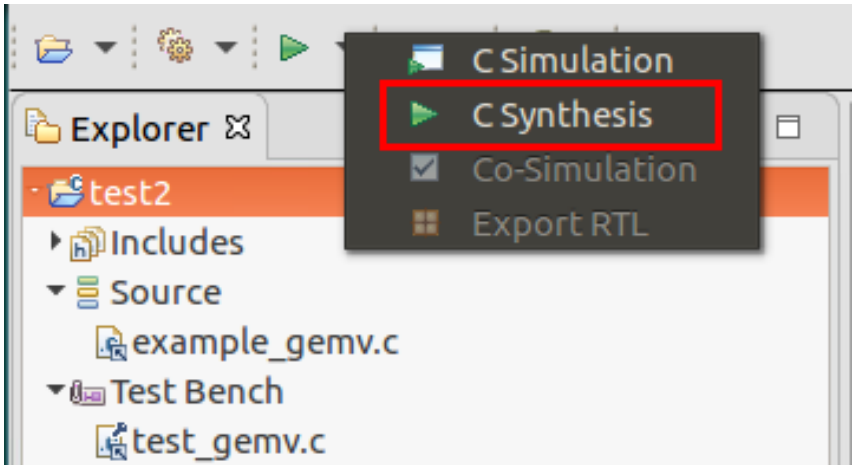
使用默认选项，点击 OK。



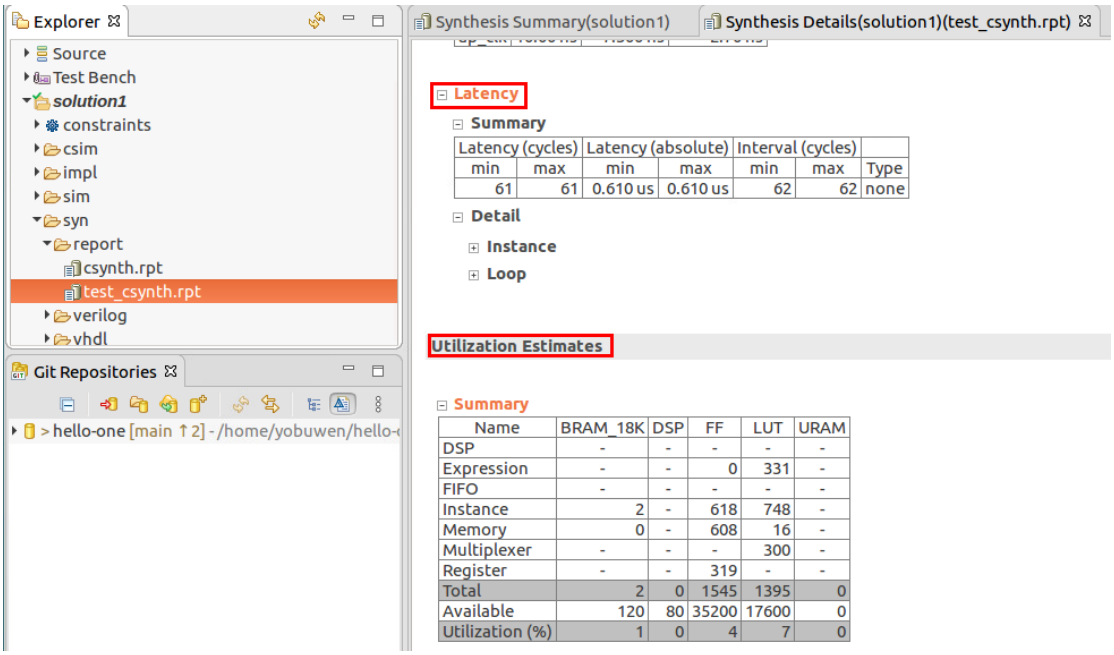
软件会打印测试结果，结果显示 C 仿真正确。



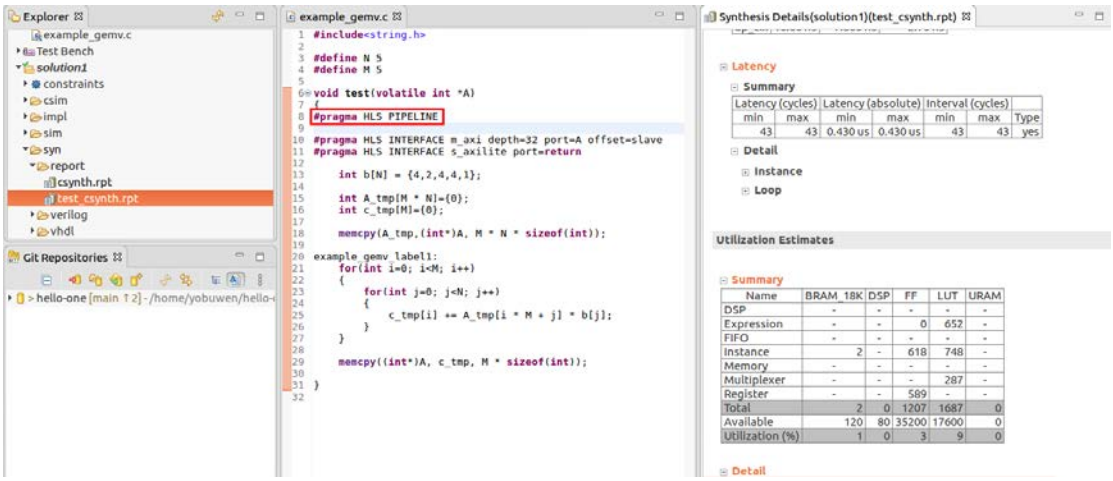
之后进行高层次综合，将 Top function 综合成对应的 RTL。



综合完成后，可在*.rpt 文件中查看综合结果（Latency 估计和资源使用情况）。



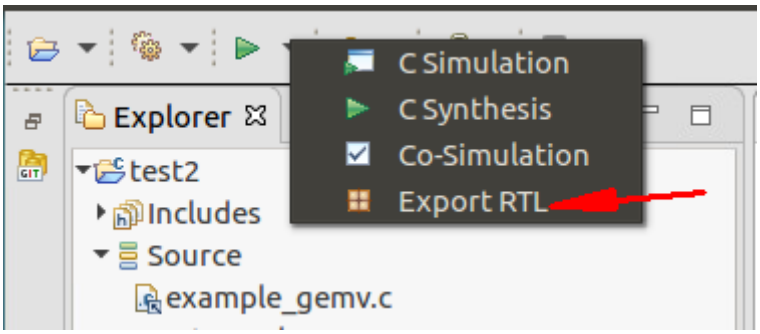
从综合结果可以发现，计算所用时间 0.610us，下面我们对计算结构进行优化，在源码中添加优化指令，再次进行高层次综合，综合结果如下图所示。



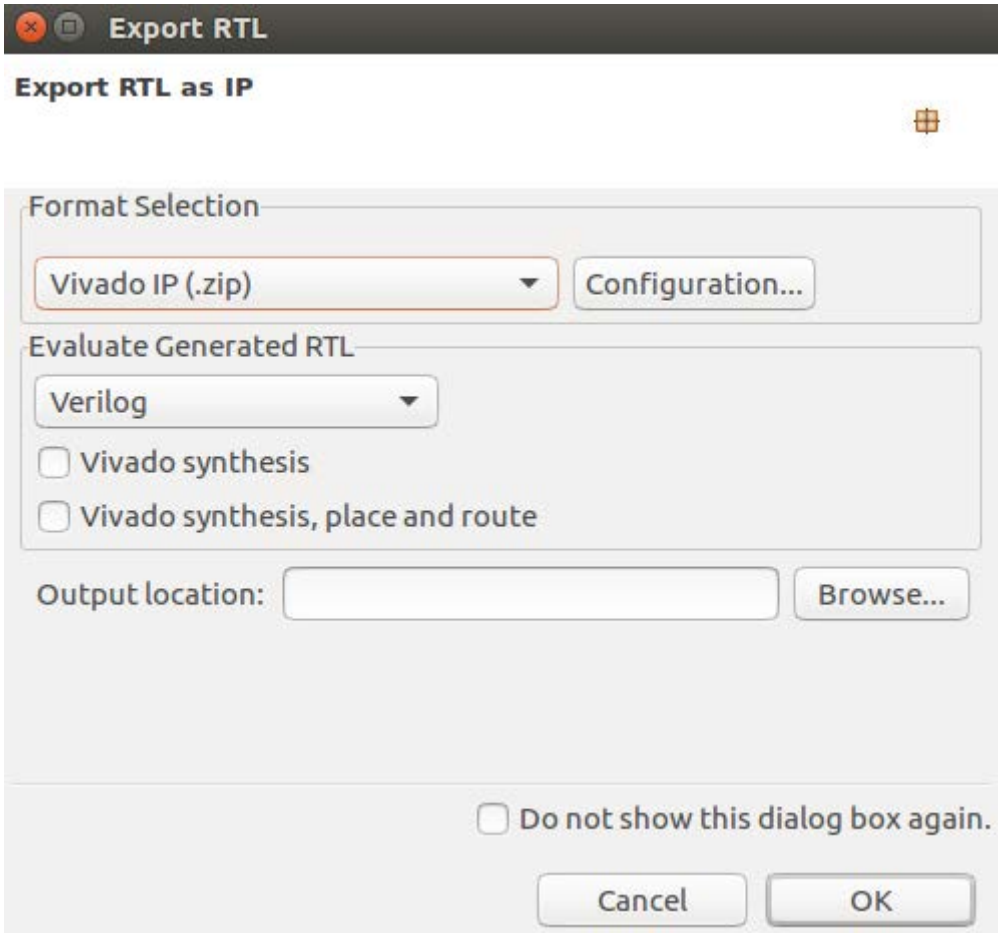
优化后计算延迟为 0.430us，计算速度提升了 1.4 倍，但相应的资源利用率也增加了。

STEP4. 数据通路搭建

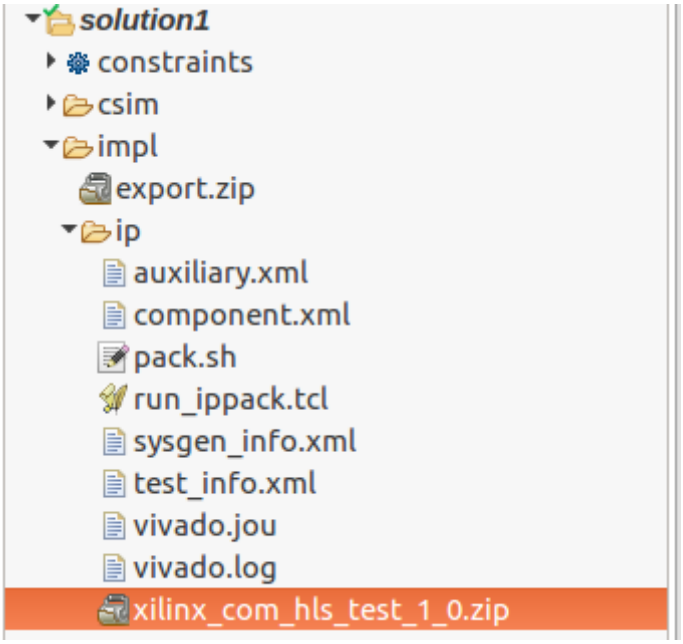
综合后，导出 RTL 设计 IP 核。



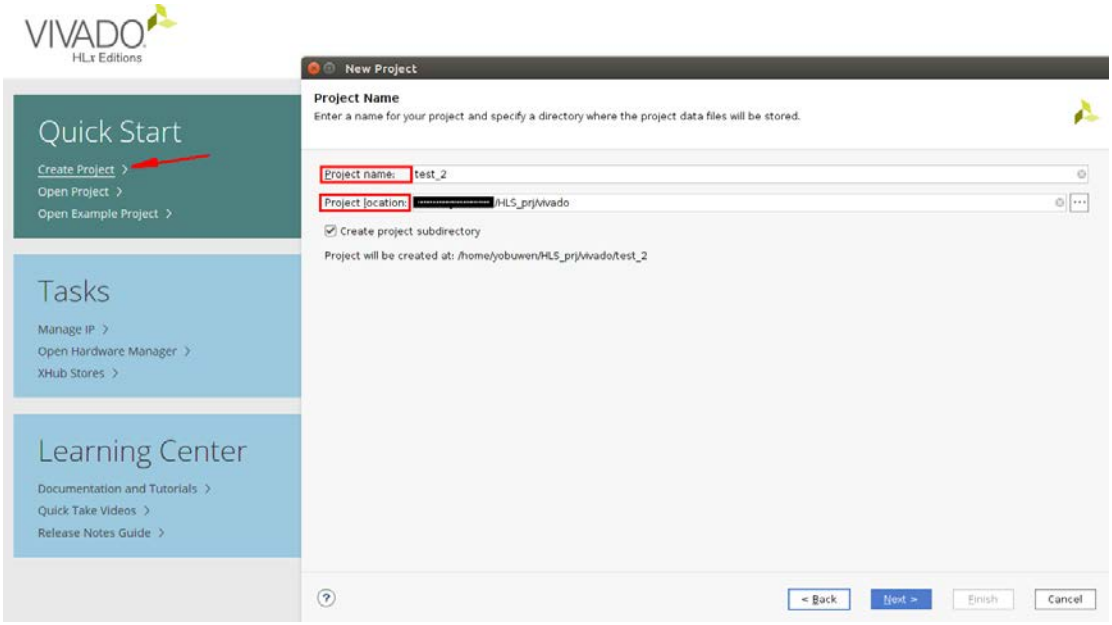
保持默认选项，点击 OK。

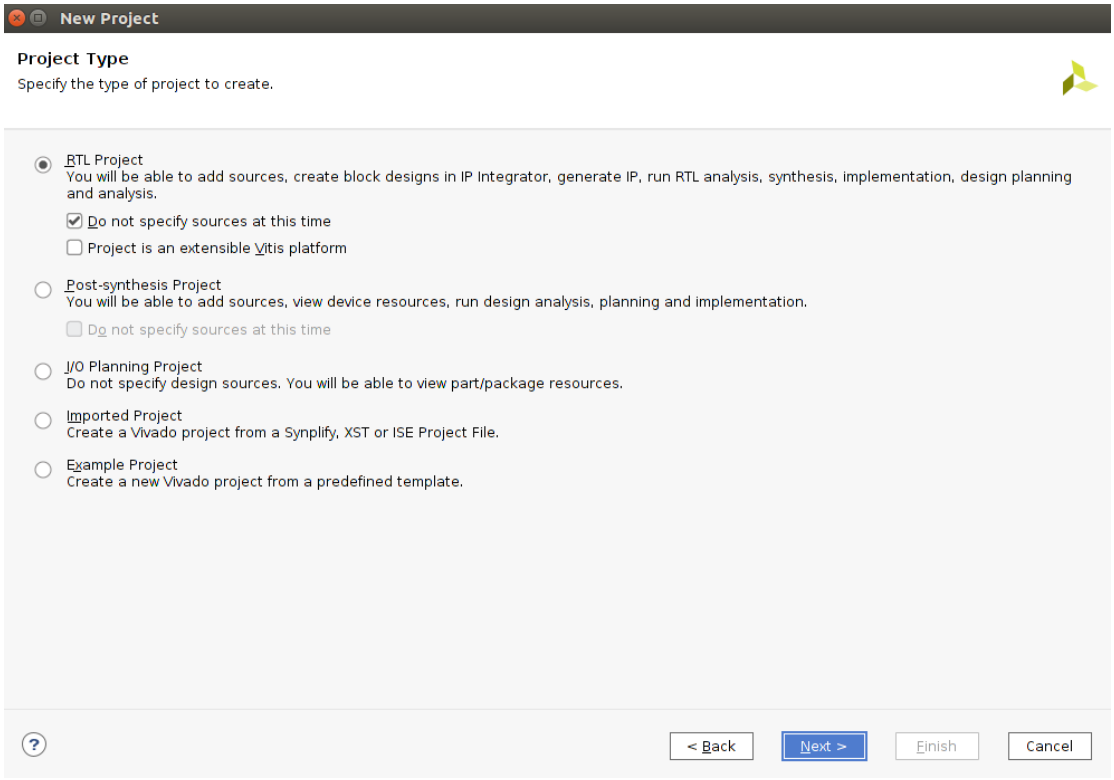


生成的 IP 核可以在 solution1---->impl---->ip 目录中找到，IP 核的文件名为 xilinx_com_hls_test_1_0.zip。

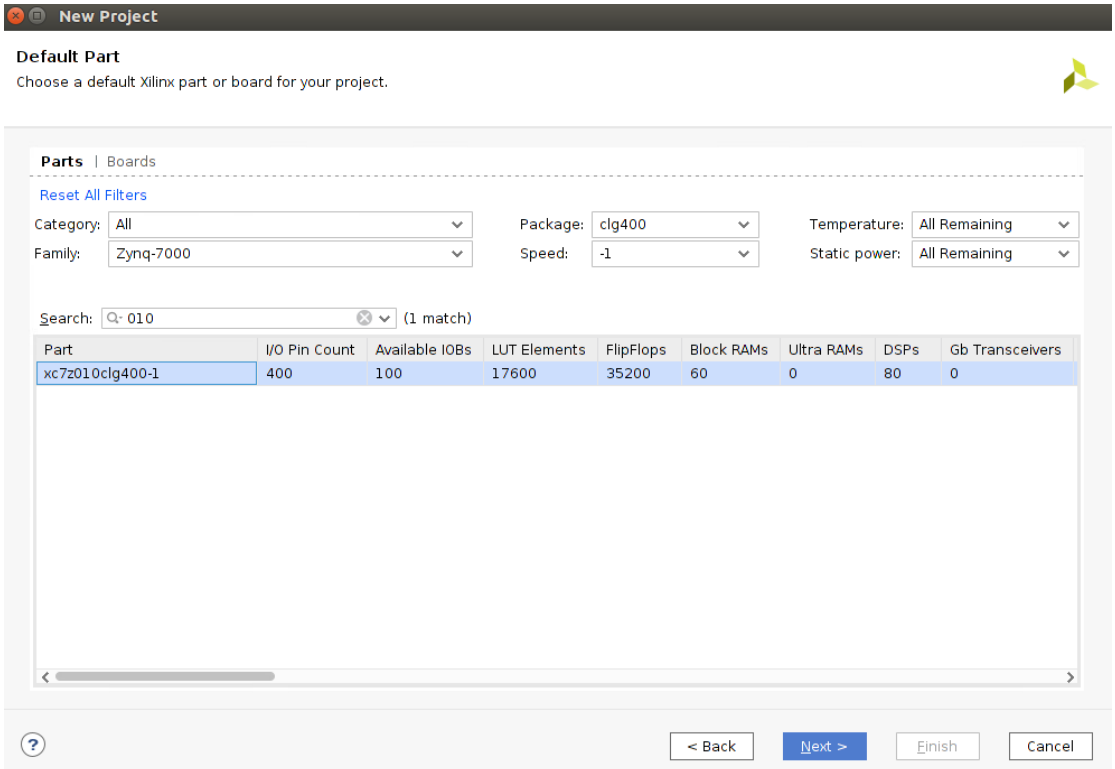


创建 Vivado 工程，首先打开 Vivado，选择 Create Project，输入工程名和工程路径。

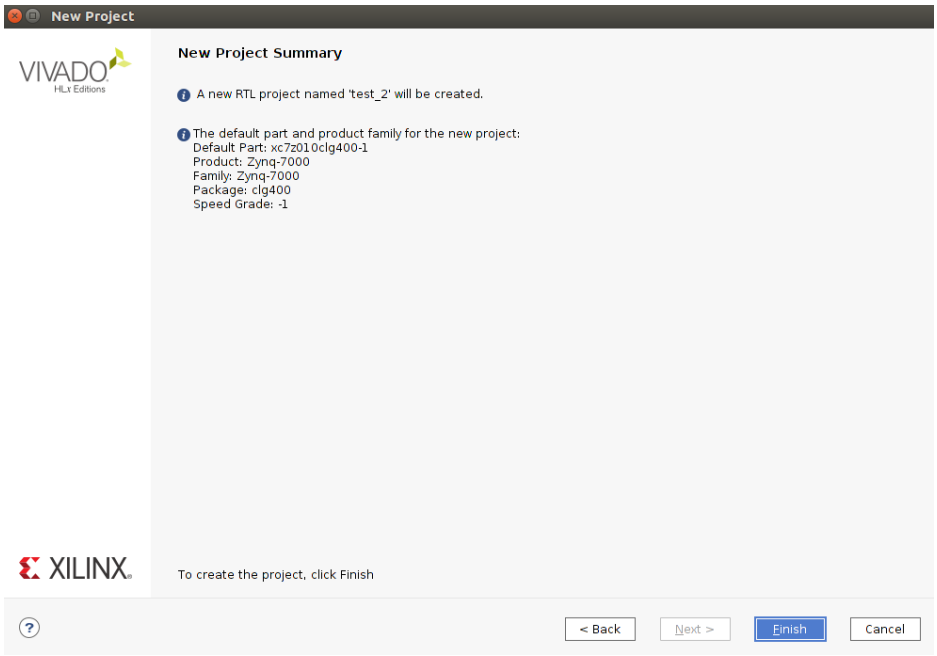




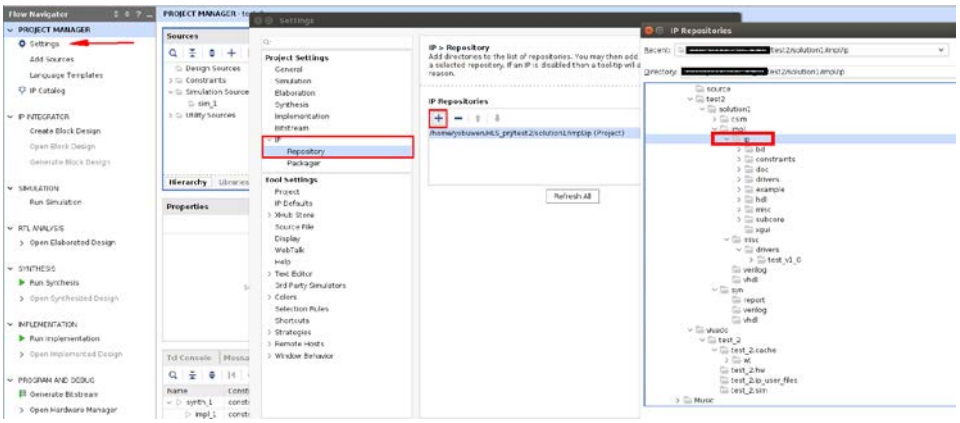
选择对应的器件设备，这里选择 xc7z010clg400-1。



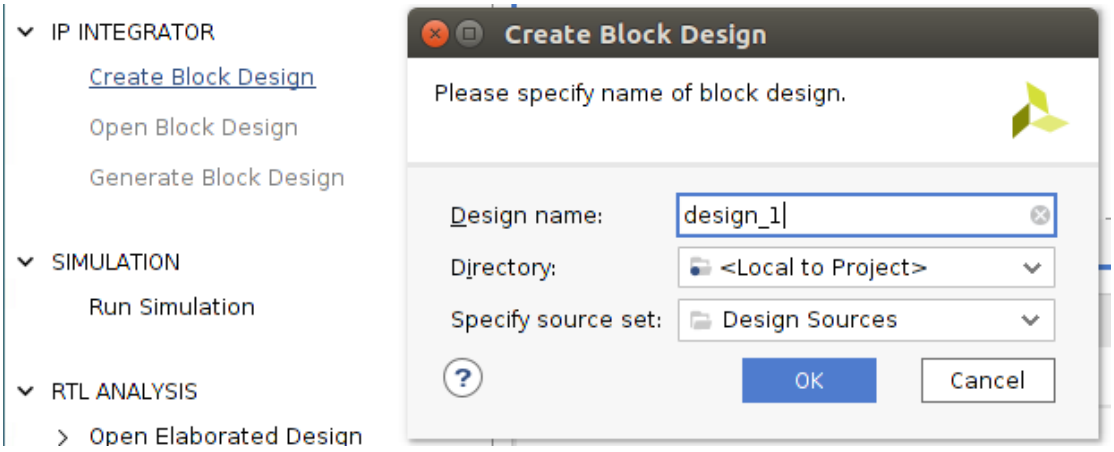
最后点击 FINISH，等待 Vivado 生成工程。



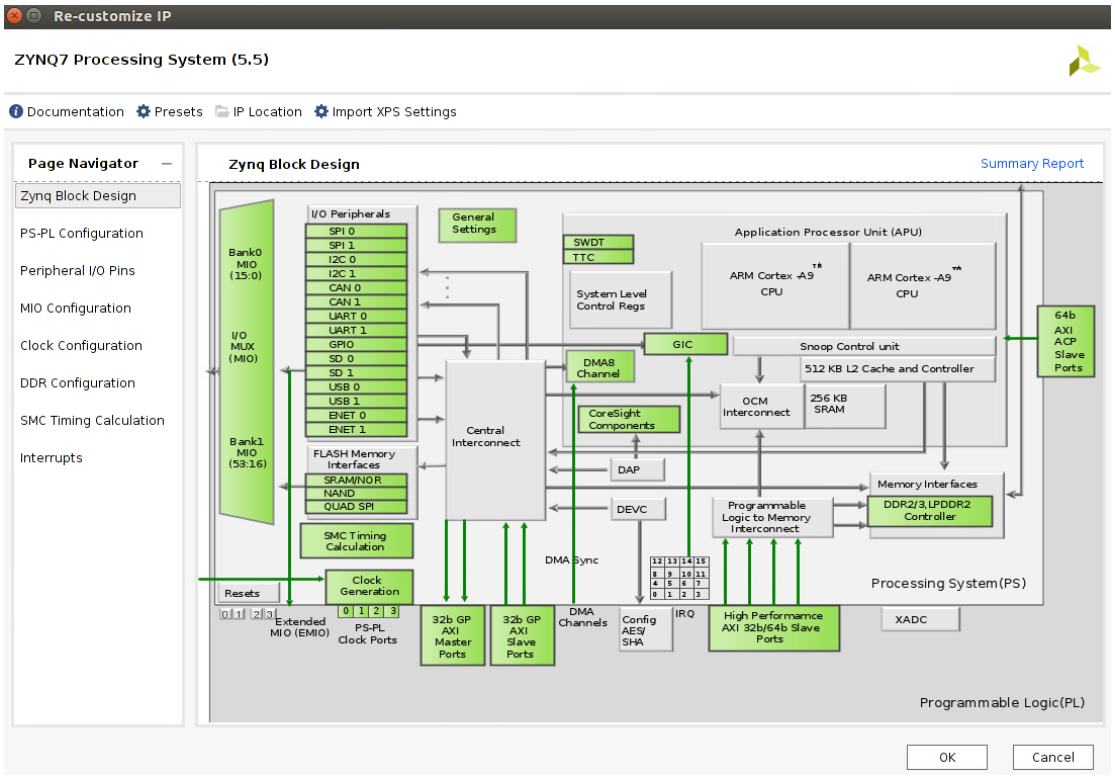
生成工程后，首先添加 HLS 生成的 IP 核。点击右侧 Flow Navigator 串口中的 Settings 选项，选择 IP---->Repository，点击加号，选择 HLS 生成 IP 核所在目录，点击 OK，会看到 IP 核已经添加到工程的 IP 仓库中。



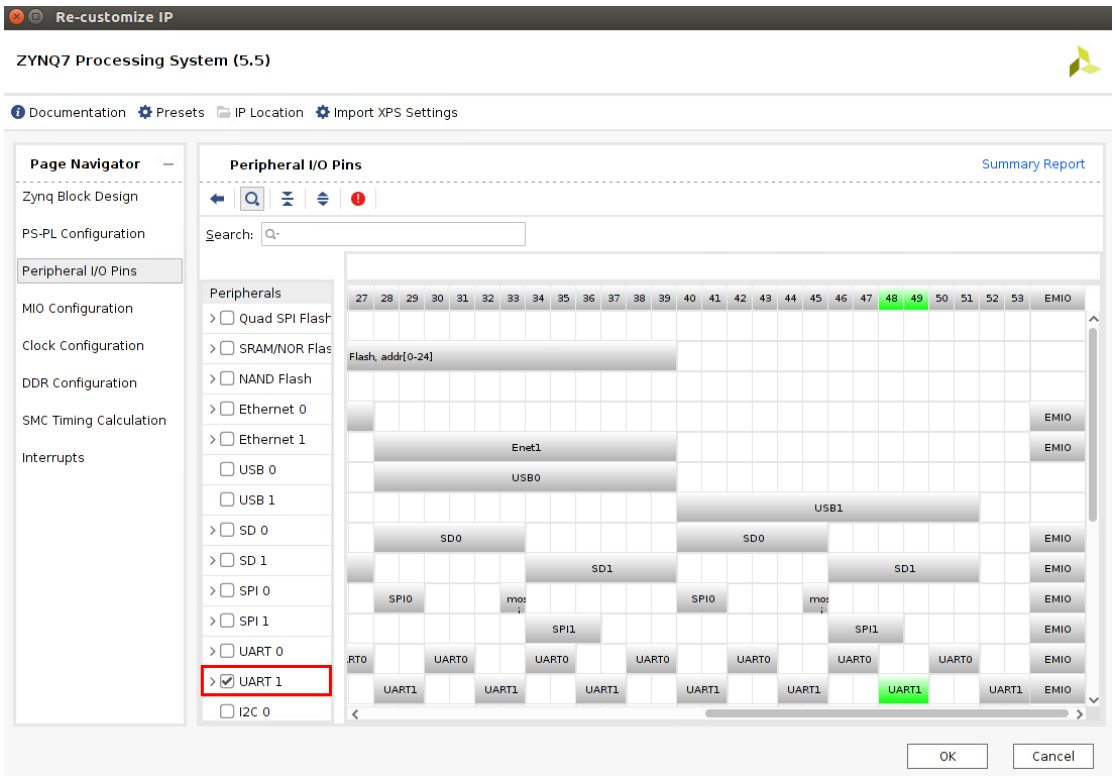
创建 Block Design，填入设计名称。



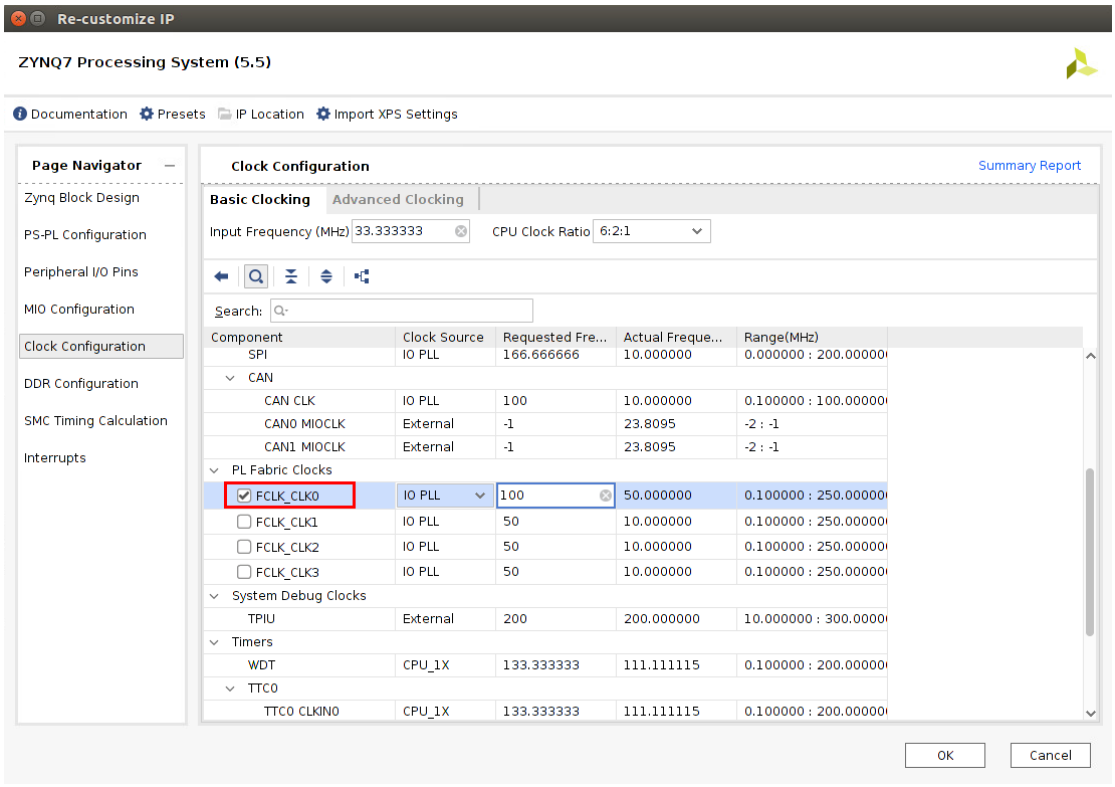
添加 HLS 综合 IP 核和 zynq processor IP 核。点击加号按钮选择添加。双击 ZYNQ IP 核对处理核心进行配置。



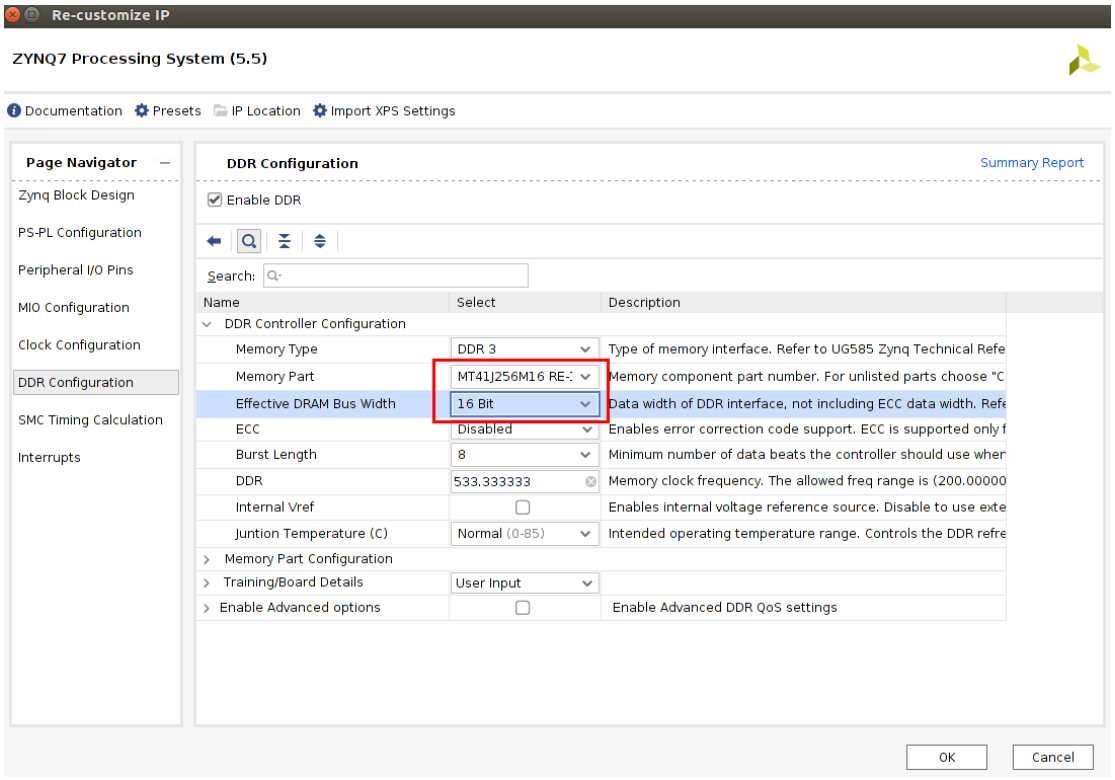
配置 UART。



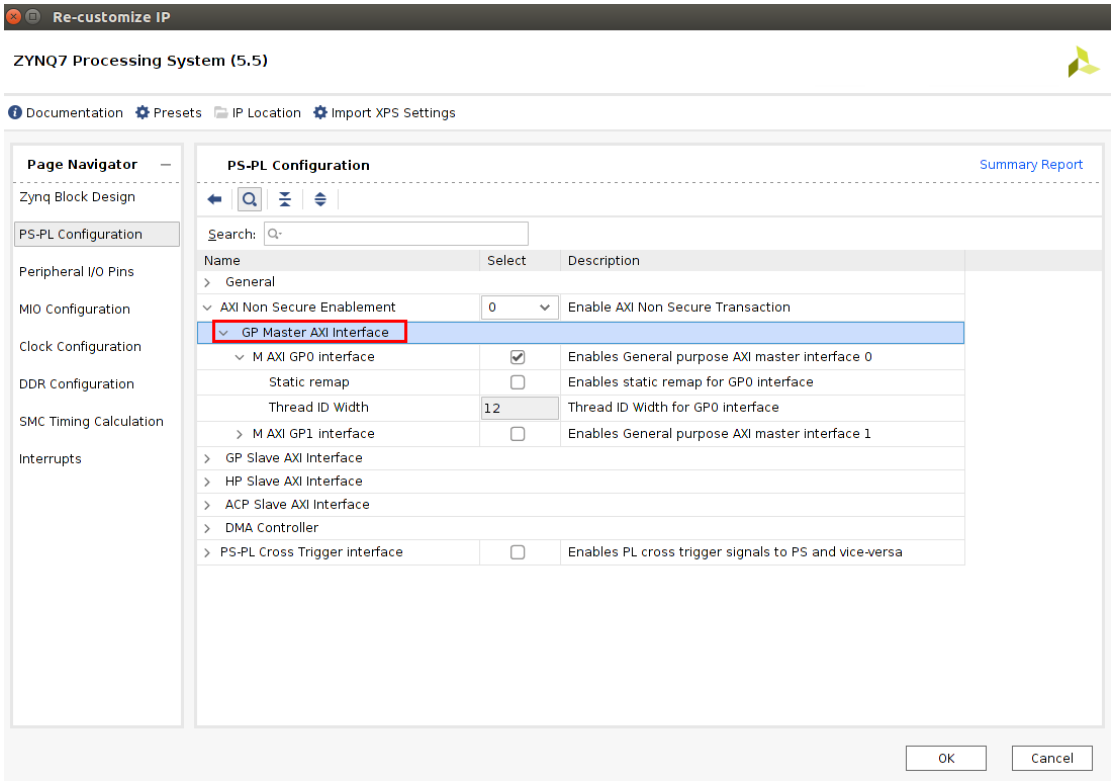
配置 PL 时钟。



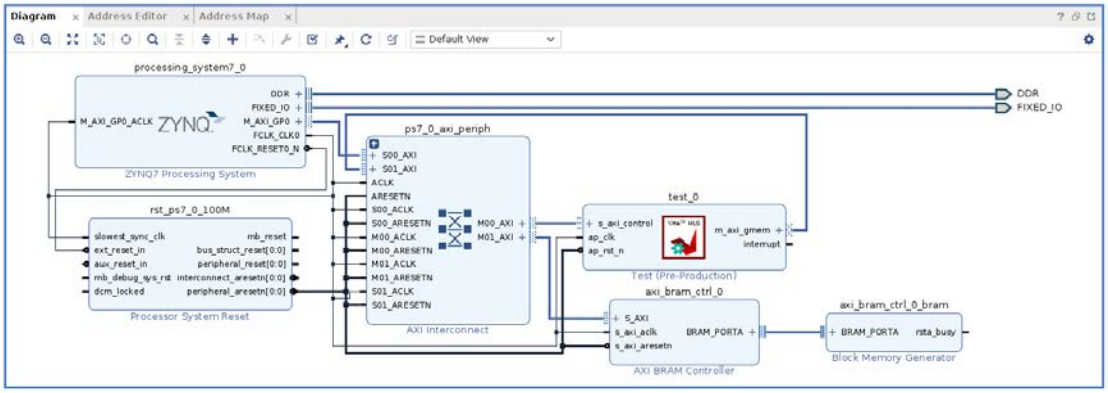
配置 DDR。



配置 AXI 总线接口。

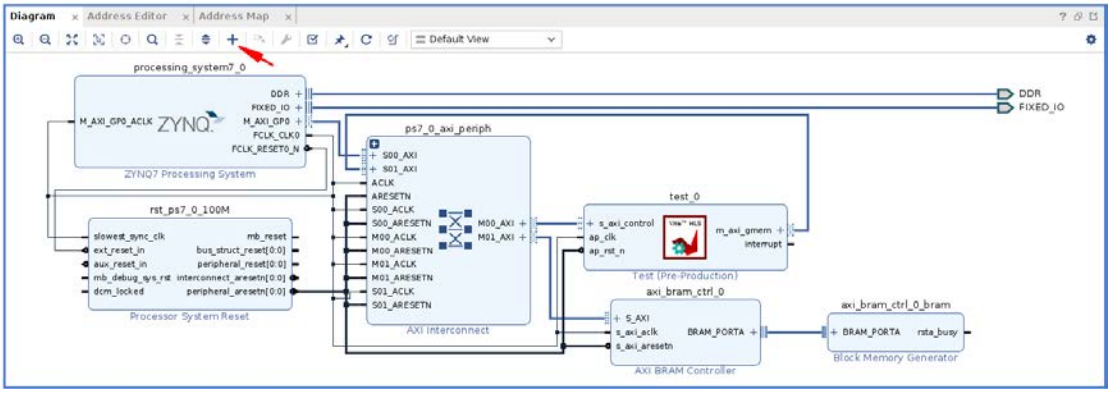


使用 Vivado 提供的自动配置和自动布线工具，对 IP 进行基础配置和连接。

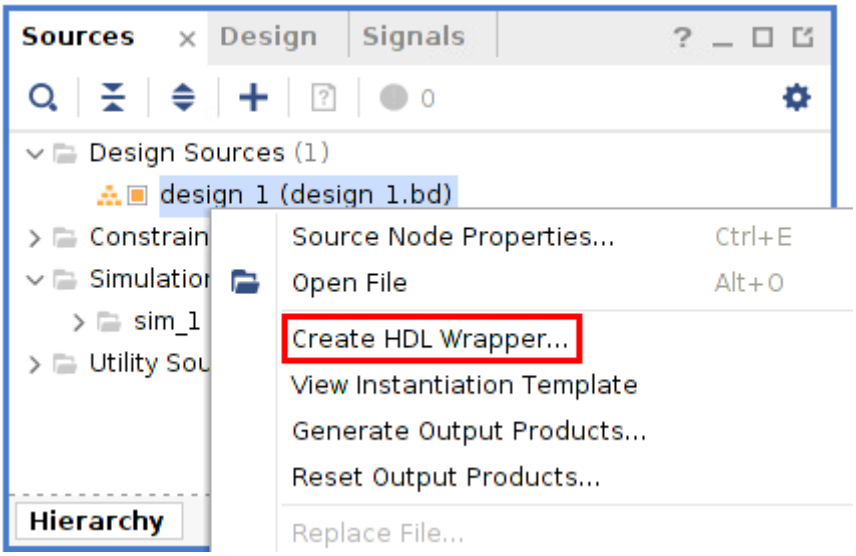


注意：设计中添加了 AXI BRAM Controller IP 核用来存储计算参数。

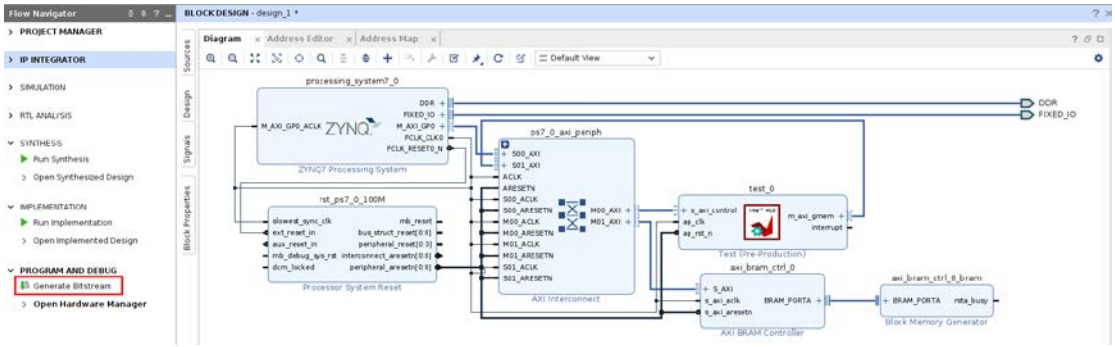
点击 validate design 验证设计的正确性。



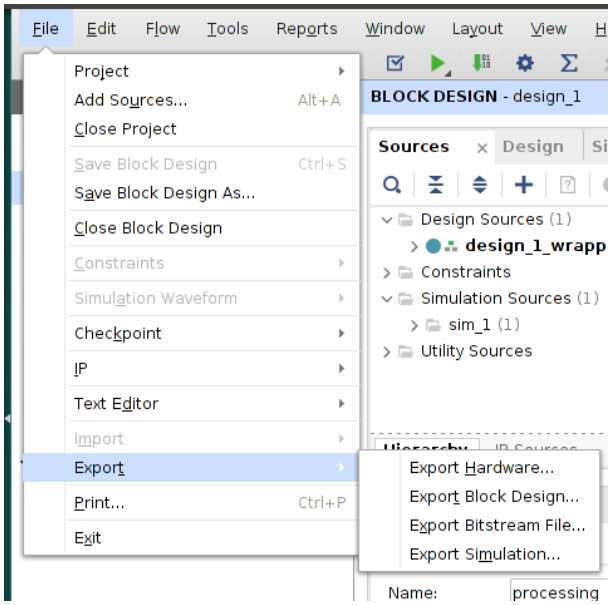
在 Source 对话框中选择设计文件，右键生成顶层设计文件（直接点击 OK）。



至此，设计配置全部完成，生成 bitstream 文件（使用默认选项）。



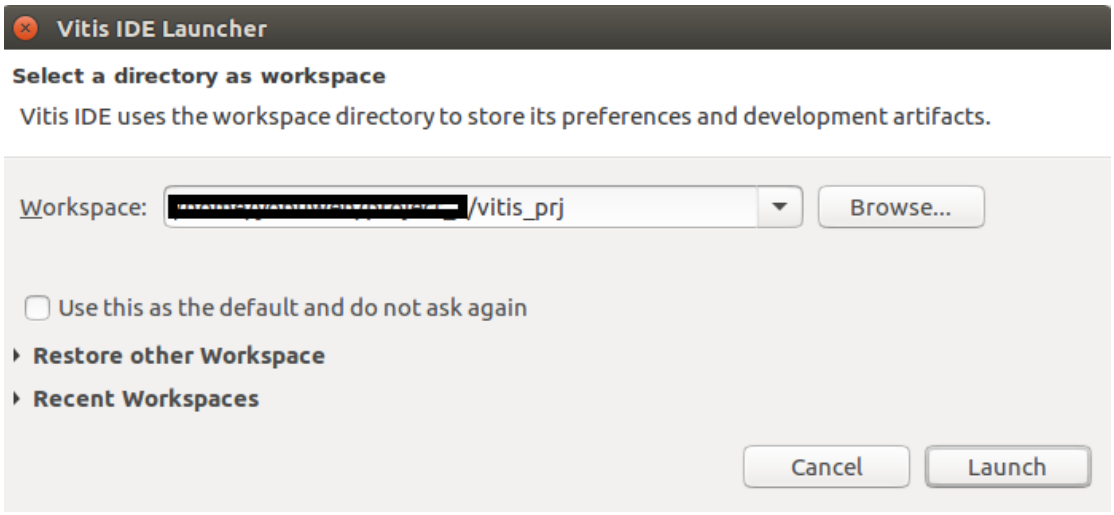
导出硬件，至此数据通路搭建完成。



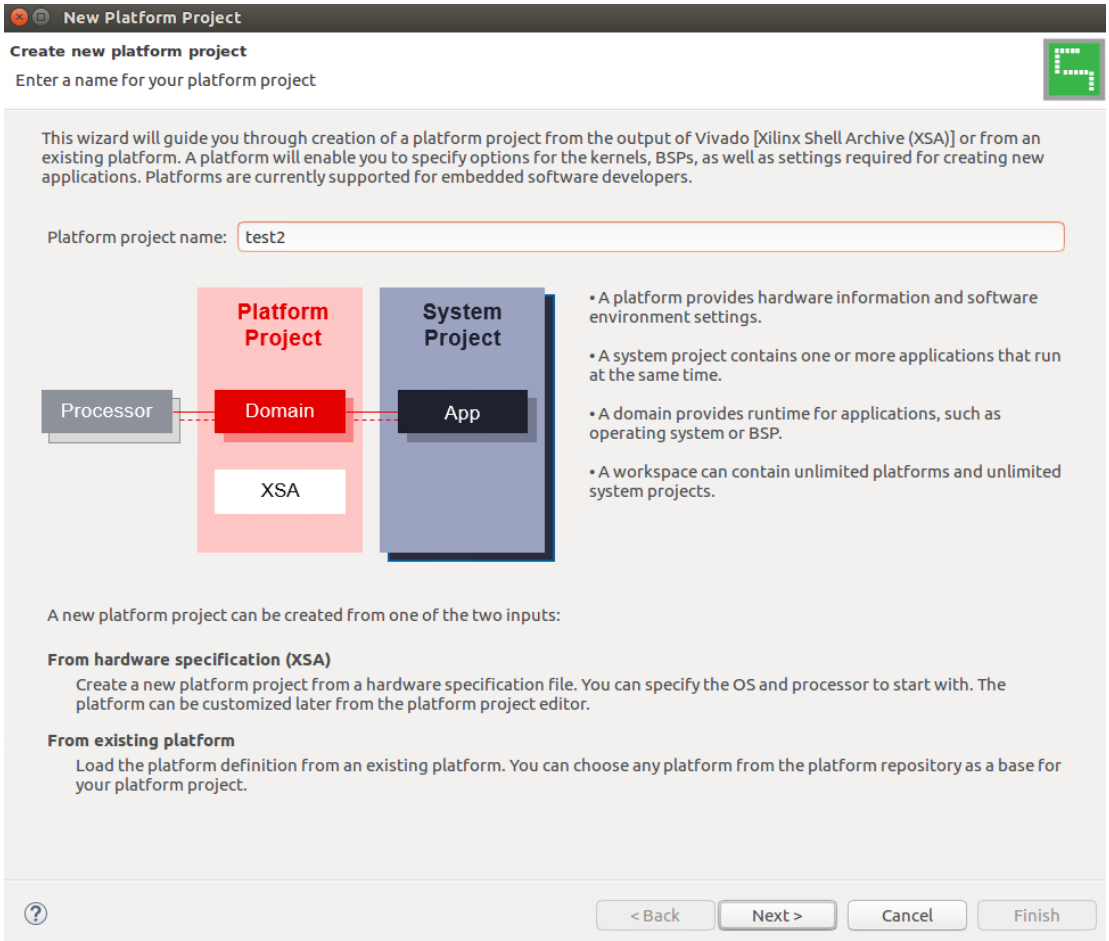
此处，不同版本软件，导出内容不同，Vivado2018 导出为.hdf 文件。Vivado2020.2 导出为.xsa 文件。导出时注意要包含 bitstream 文件。

STEP5. 处理系统设计

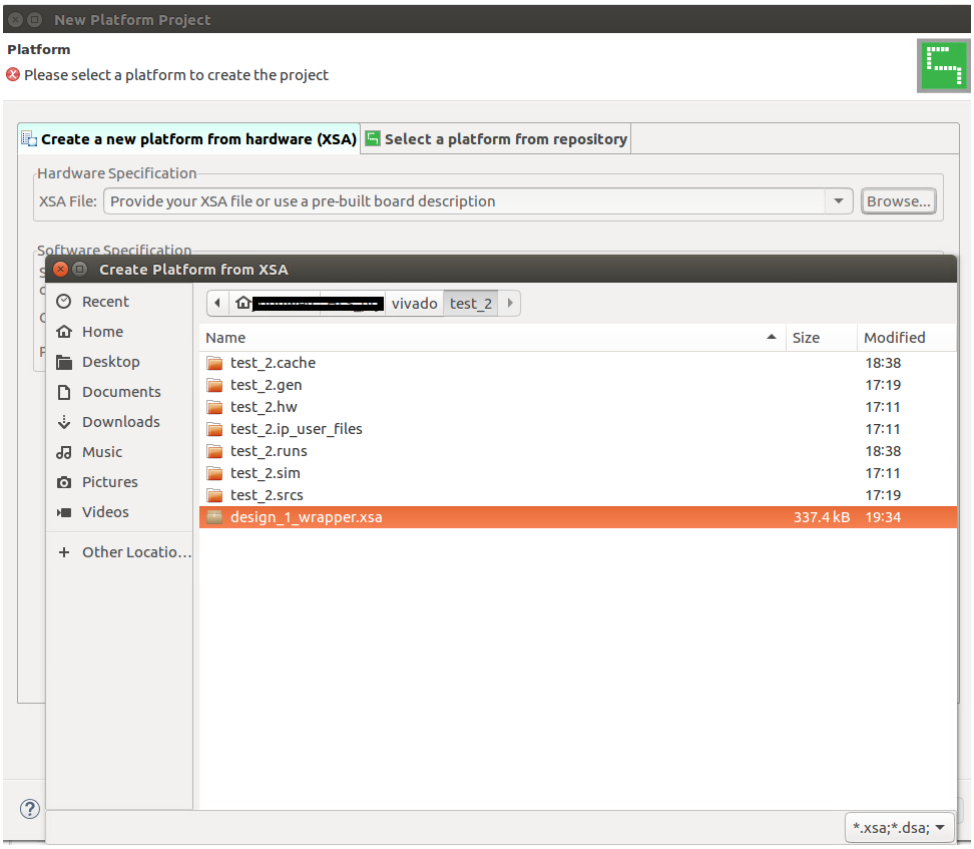
打开软件开发环境，Vitis（SDK），首先穿件设计空间。



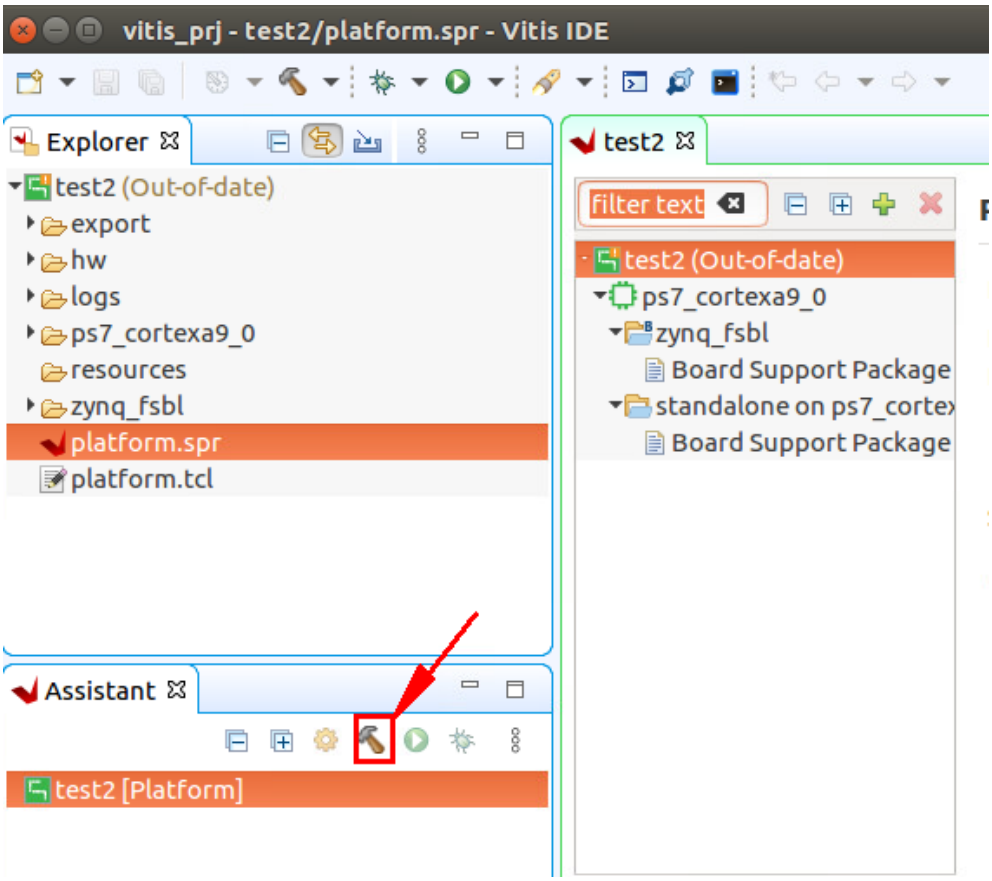
基于硬件设计文件，创建 Platform 工程，输入工程名。



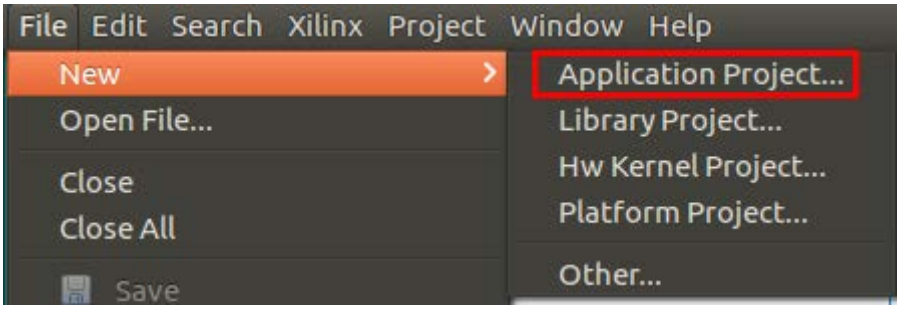
选择生成的 xsa 文件，之后点击完成。



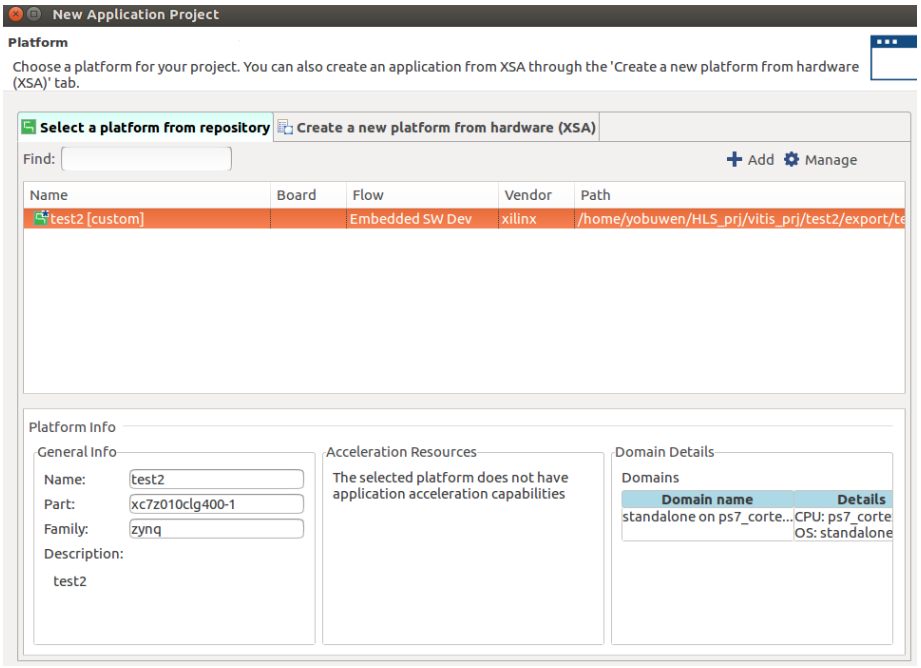
Vitis 自动生成硬件平台工程，对工程进行编译。



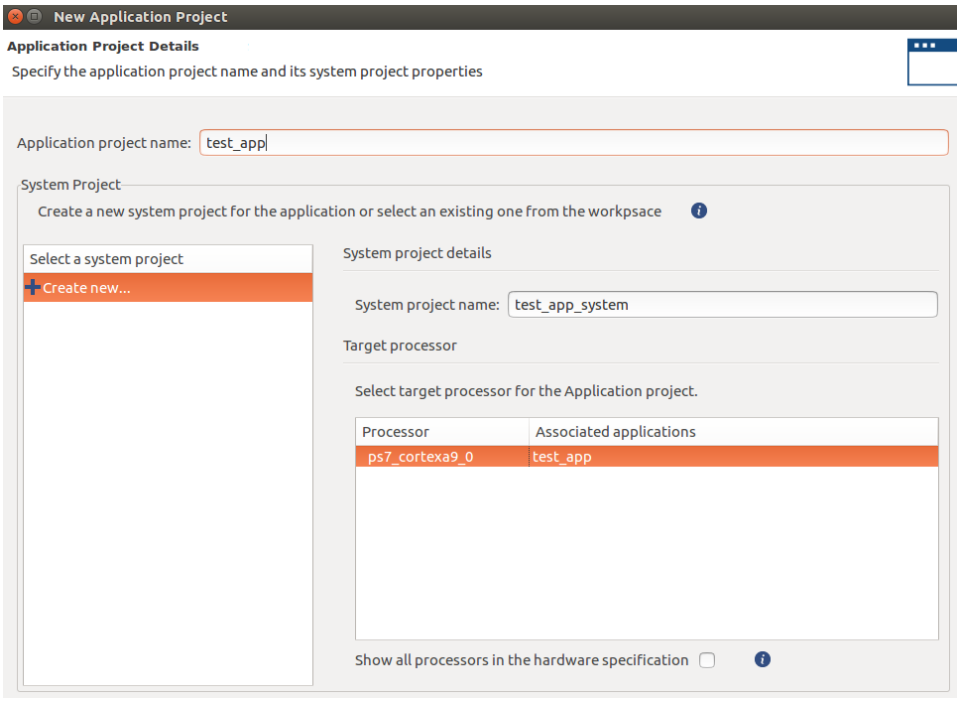
编译成功后，创建 Application 工程。



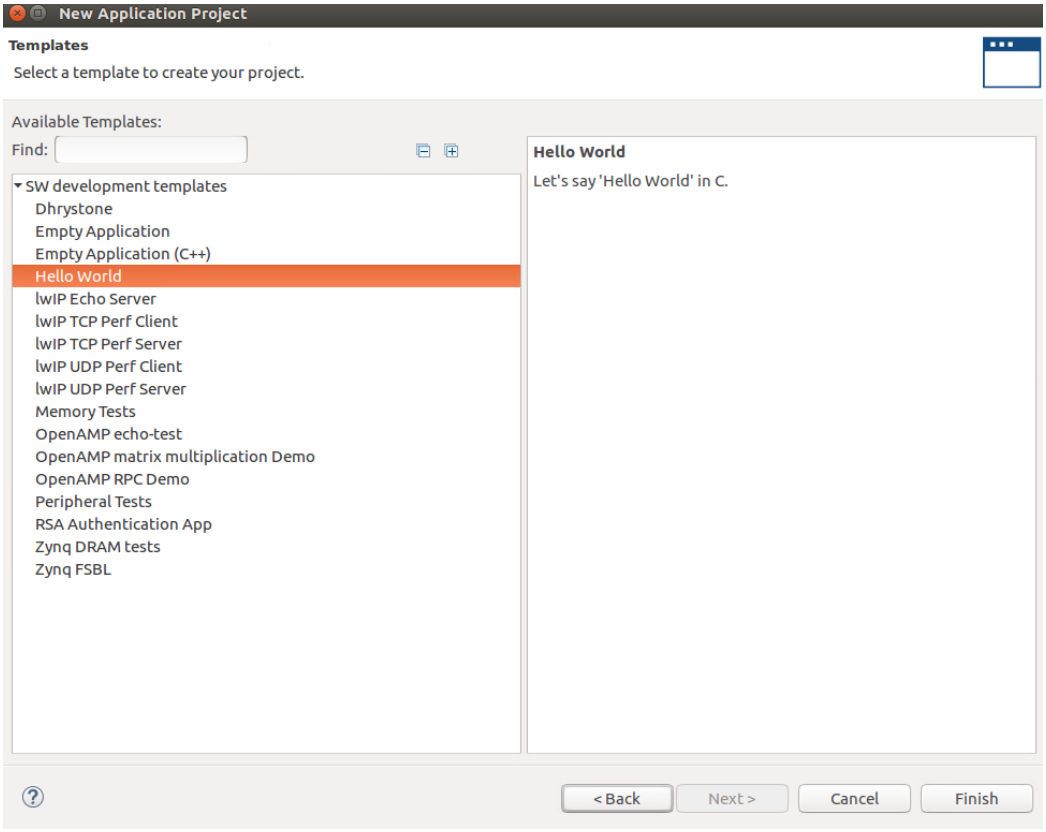
选择之前编译的硬件平台。



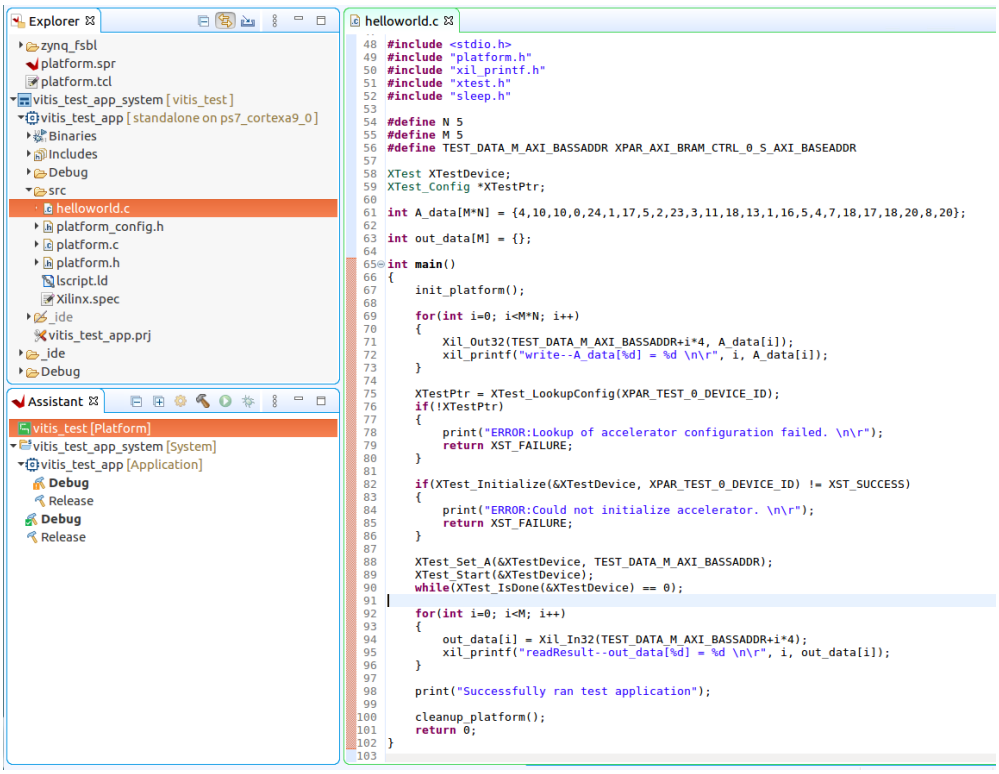
输入工程名。



点击 NEXT，直至选择工程模板。此处选择 Hello World 模板建立工程。



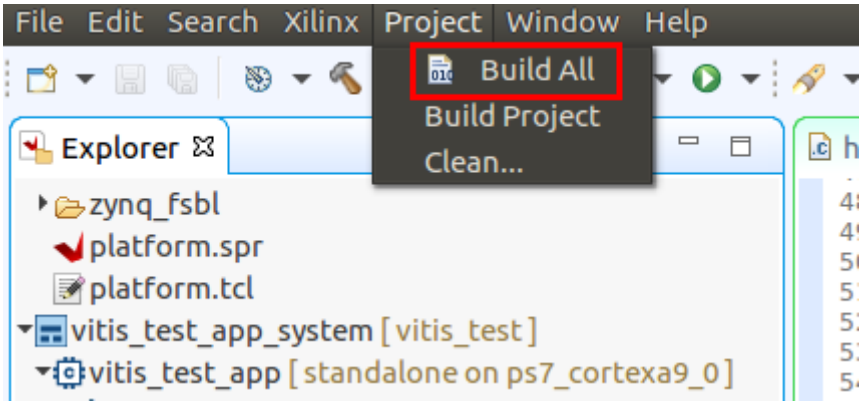
等待 Vitis 自动建立软件工程后，打开 helloworld.c 主函数文件，输入设计文件。



C 代码原理：首先将测试数据循环写入 BRAM 中，之后执行 Test IP 的初始

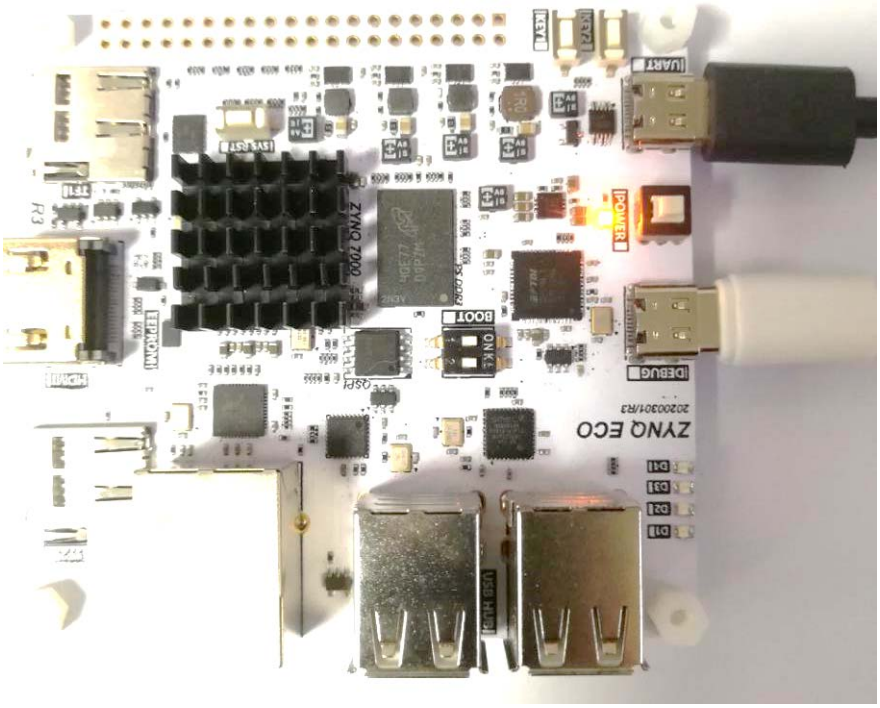
化程序，并使用 XTest_Set_A()函数将 BRAM 存储参数传入 IP 核。准备好计算所需参数后，使用 XTest_Start()启动 IP 计算，待 IP 核计算完成后，再从 BRAM 地址中读取计算结果验证过程正确性。

对设计工程进行编译，点击菜单栏中 Prject---->Build All.

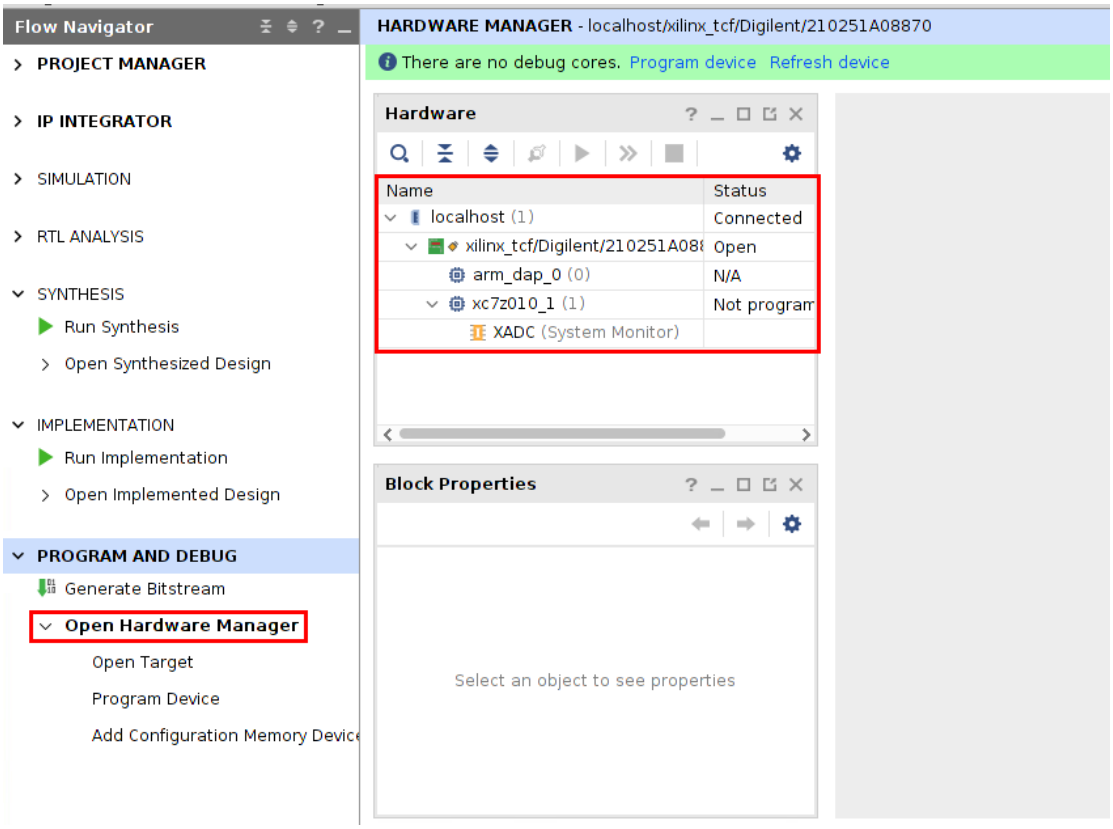


STEP6. 上板验证

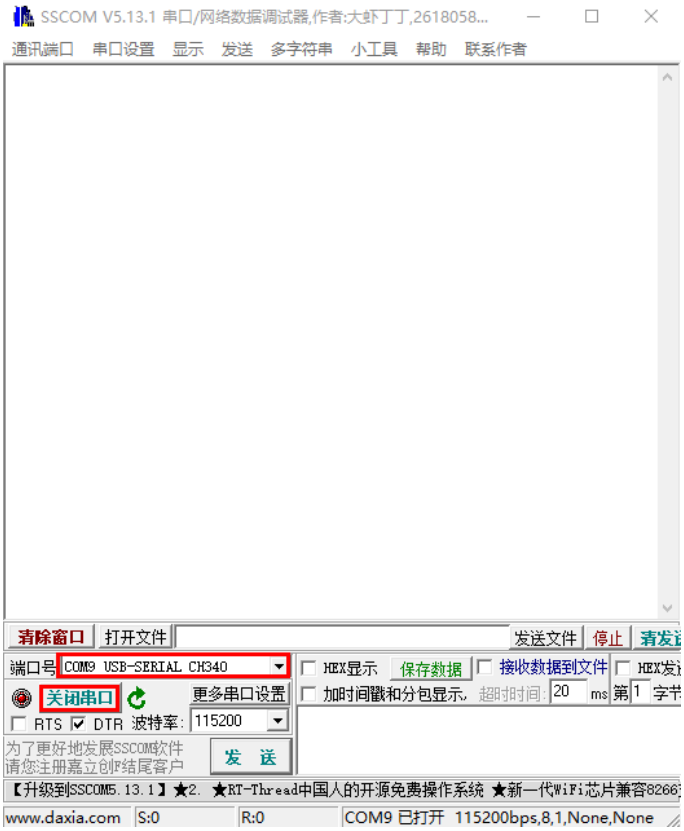
连接 Zynq 7010 板卡的 Jtag 和 UART 接口，打开电源开关，如下图所示。



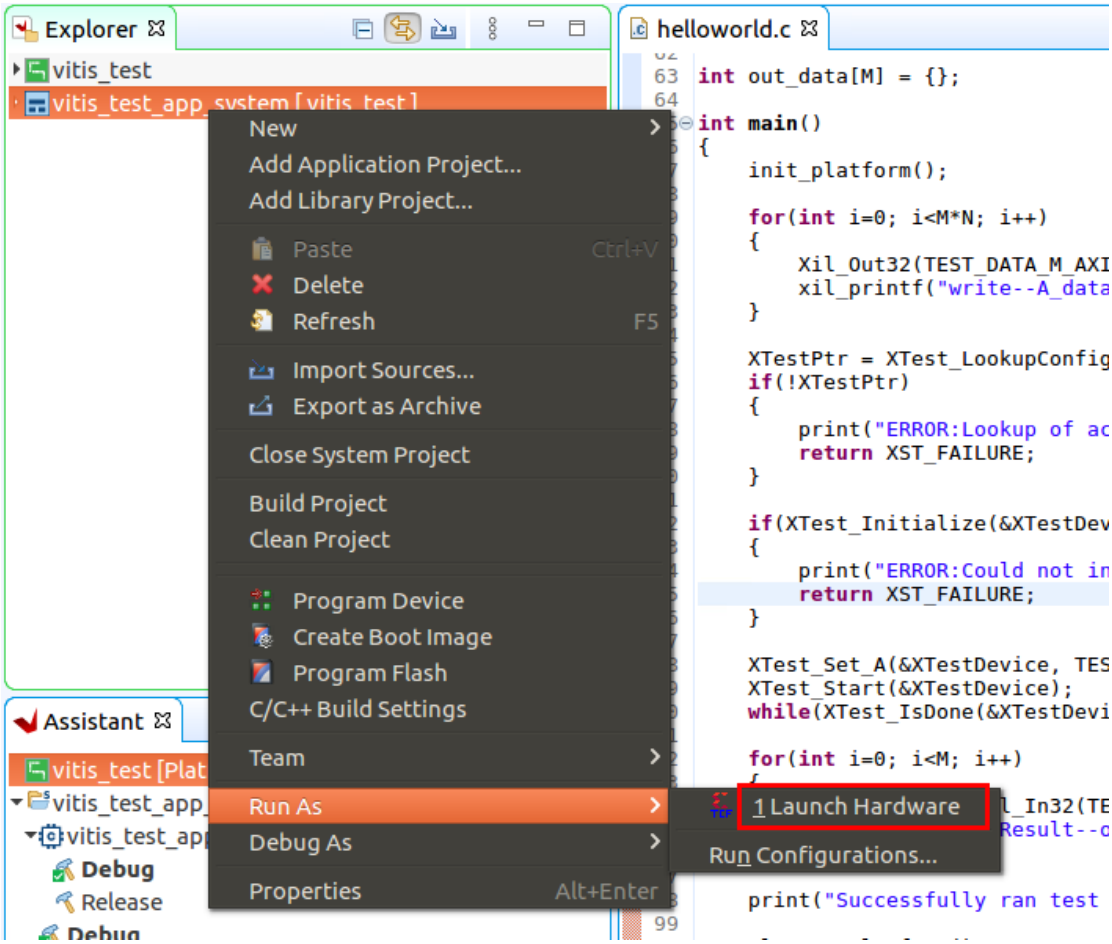
使用 Vivado 软件连接板卡。



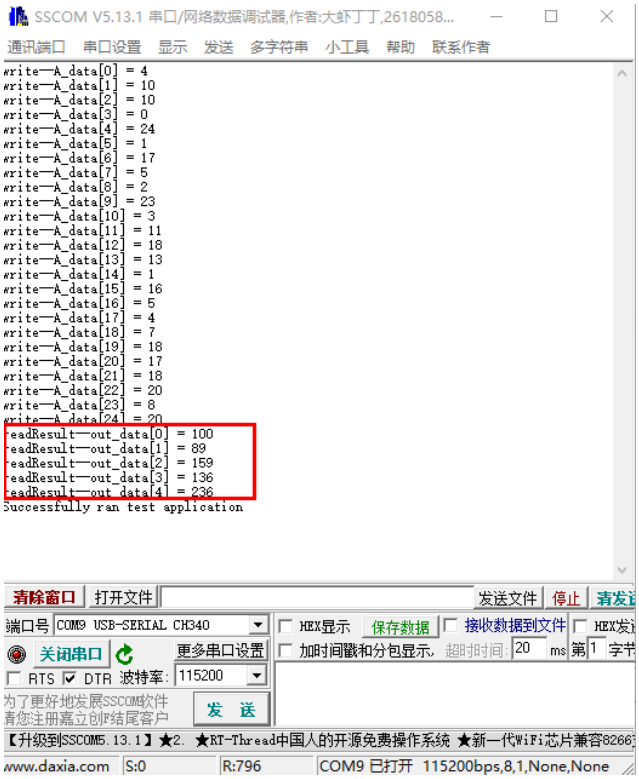
在 PC 端打开串口调试助手，连接板卡 UART 端口。



连接成功，并自动识别芯片型号正确后，使用 Vitis 软件向板卡烧写编译好的程序。选择工程文件，右键 Run As---->1 Launch Hardware，等待程序烧写完成。



等待程序烧写成功后，在串口调试助手中打印结果信息，如下图所示。



程序运行成功，计算结果正确。

1.7 实验任务

采用 C/C++语言，设计脉动阵列结构，利用 Vitis HLS（或 Vivado HLS）工具进行仿真和综合，并输出 RTL IP 核。

基于 Zynq7010 开发板搭建数据通路，集成脉动阵列 IP，实现嵌入式计算系统，并完成上板验证。

最终形成实验报告，包括实验目的、实验原理、实验结果、结果分析、C/C++代码等。

附录

Vitis 2020.2 下载链接：
https://china.xilinx.com/support/download/index.html/content/xilinx/zh/downloadNav/vitis/2020-2.html

串口调试助手下载链接：
http://www.daxia.com/sscom/sscom5.13.1.rar