Use Vivado to build an Embedded System

引言

在这个实验里，你将使用Vivado创建一个面向ZYBO板的基于ARM Cortex-A9处理器的简单设计。你将使用Vivado构建硬件系统，并用SDK(Software Development Kit)来搭建一个用例应用程序以验证硬件功能。

目的

完成本实验后，你将能够：

* 为Zynq体系创建Vivado工程
* 用IP集成器创建硬件系统
* 用SDK创建标准内存测试工程
* 在板子上运行测试程序

步骤

这个实验分为若干个步骤，每个步骤都会包含概览性描述，这些描述为接下来的详细指示提供了信息。根据详细指示来完成实验。

实验包含五个基本步骤：你将用Vivado创建一个顶层工程，用Vivado IP Integrator 创建处理器体系，生成顶层**Verilog HDL** 并导出设计到SDK，在SDK里创建内存测试程序，最后在硬件上进行测试。

设计描述

实验的目的是让你了解一个完整的软硬件处理器体系设计流程。每个实验都基于之前的实验。下面的示意图表示了完整的设计。 (**Figure 1**).

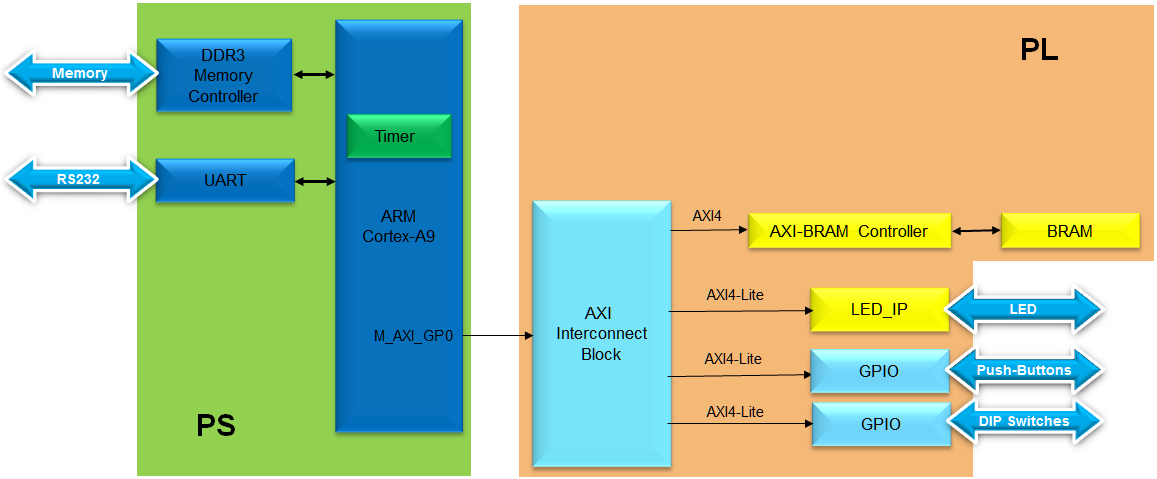


Figure 1. Completed Design

本实验中，你将使用IP Integrator 来创建一个包含下面部分的基于处理系统的设计 (**Figure 2**):

* ARM Cortex A9 核(PS)
* UART 用于串行通信
* DDR3 控制器 用于 DDR3\_SDRAM 内存

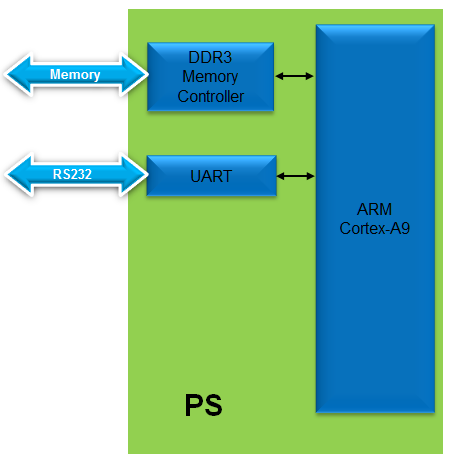


Figure 2. Processor Design of this Lab

本实验总体流程

Step 5:

Verify Functionality in Hardware

Step 4:

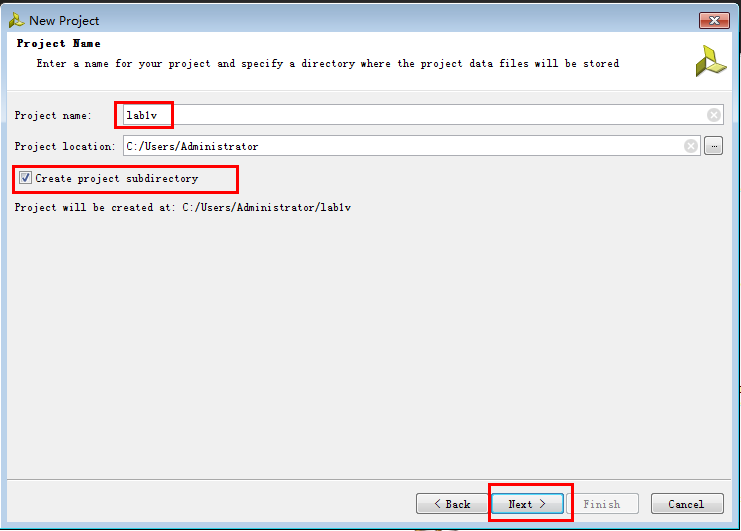
Create a Memory TestApp in SDK

Step 3: Generate Top-Level HDL and Export to SDK

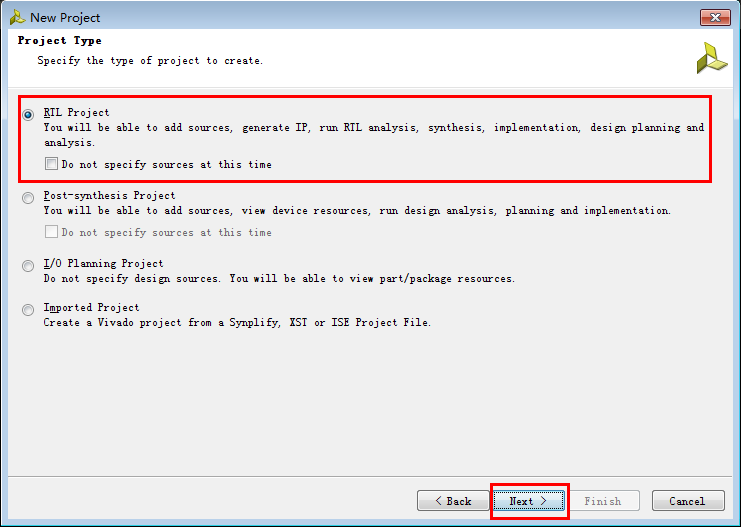
Step 2: Create Processor System using IP Integrator

Step 1: Create a project using Vivado

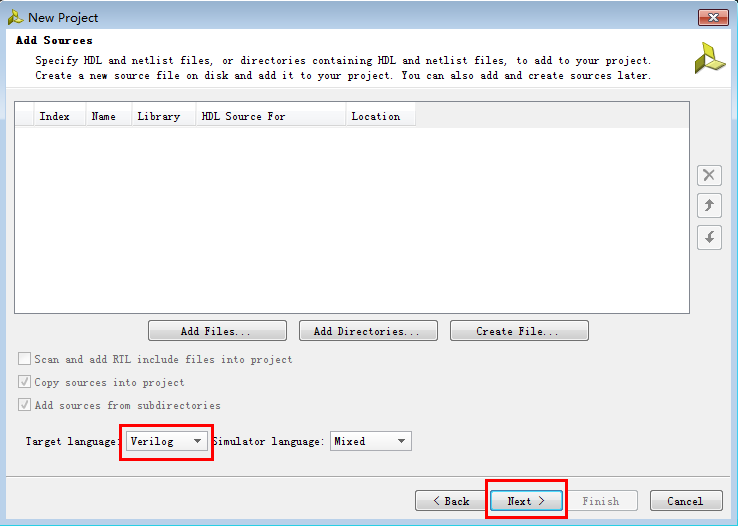
1. 创建一个Vivado 工程 Step 1
   1. 启动 Vivado ，创建一个面向ZYBO的空工程(使用 xc7z010clg400-1 设备) 并使用Verilog HDL.
      1. 打开 **Vivado 2013.4。**
      2. 选择 **Create New Project。**
      3. 如下图，将工程命名为lab1v，并选择合适的工程路径。勾选Create Project Subdirectory，点击Next。



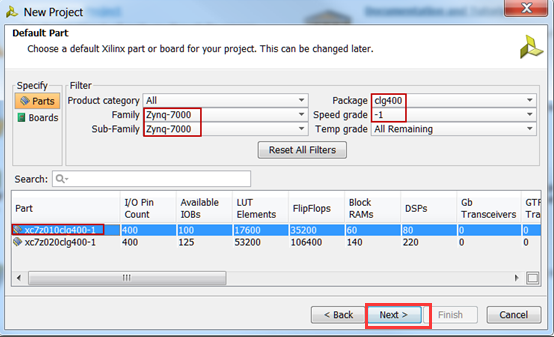
* + 1. 选择RTL Project选项，点击Next.



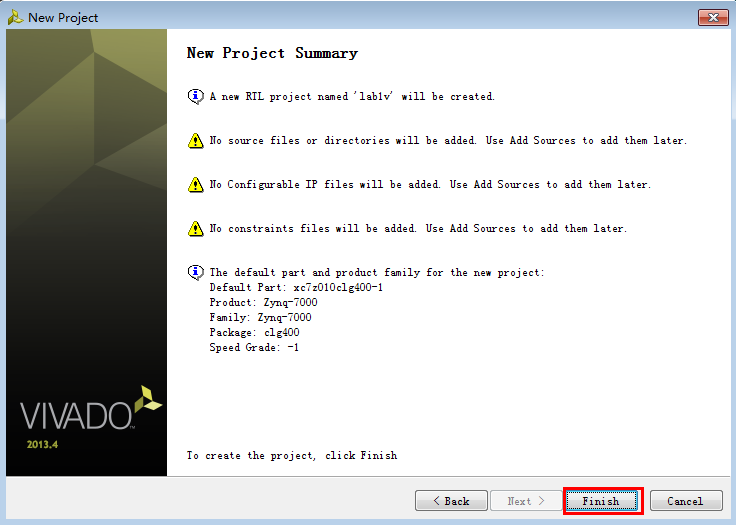
* + 1. 选择**Verilog** 作为目标语言， Mixed 作为模拟器语言，点击 Next.



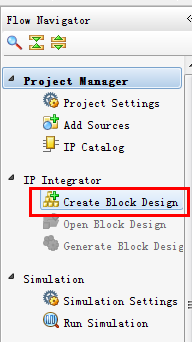
* + 1. 再点击两次 Next，跳过 Adding Existing IP 和Add Constraints步骤。
    2. 按下图进行设置。注意要选择**xc7z010clg400-1** 部分，因为它是ZYBO板上的。点击Next。



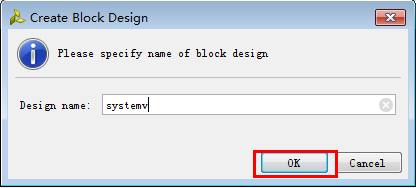
* + 1. 点击 Finish ，创建一个空的Vivado 工程。



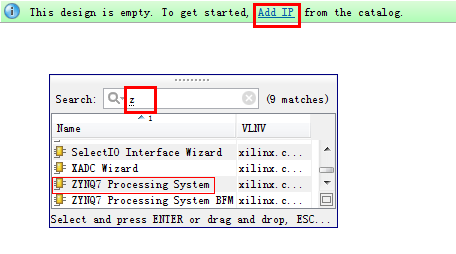
1. 使用 IP Integrator创建系统 Step 2
   1. 使用 IP Integrator 来创建一个新的模块设计，添加ZYNQ处理系统模块。（不导入已为ZYBO板设计好的xml文件）
      1. 在 Flow Navigator面板里, 点击IP Integrator选项下的Create Block Design。



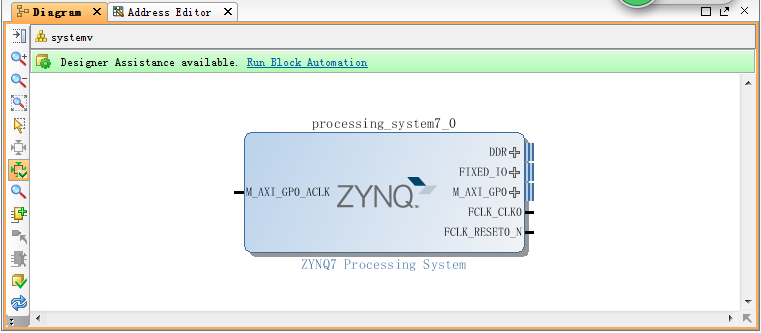
* + 1. 将设计命名为systemv，点击OK。



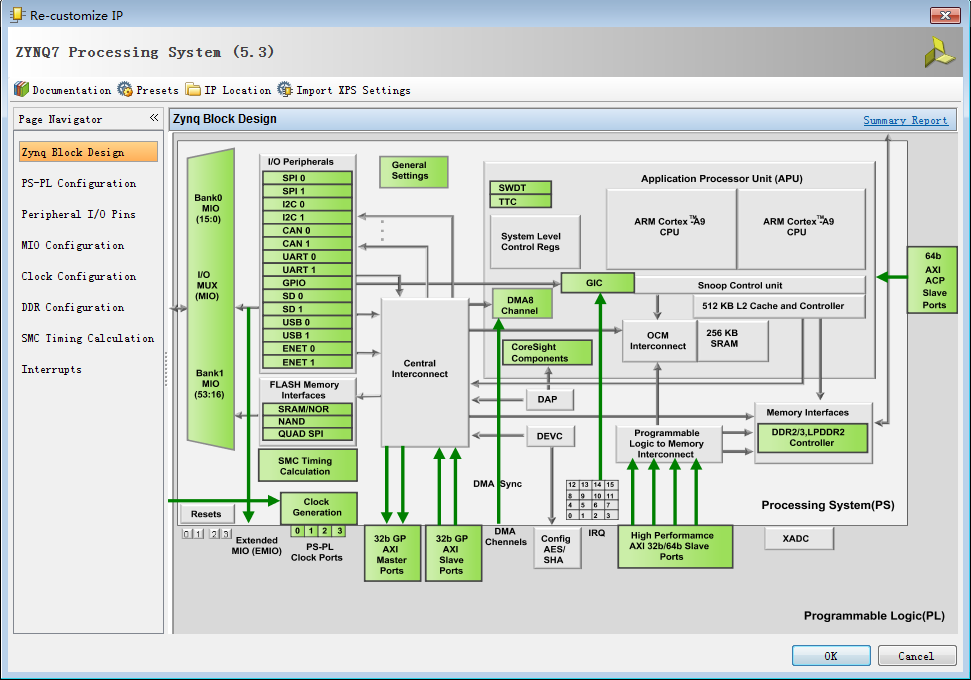
* + 1. 点击Diagram面板顶部的提示消息里的 Add IP ，在搜索栏里输入z，找到并双击ZYNQ7 Processing System这一项来将它添加到设计中。



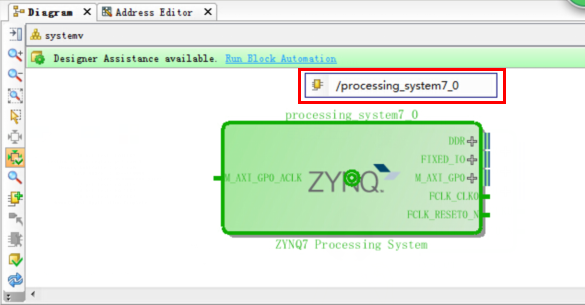
* + 1. 现在，我们得到了一个预定义好的设计模块。



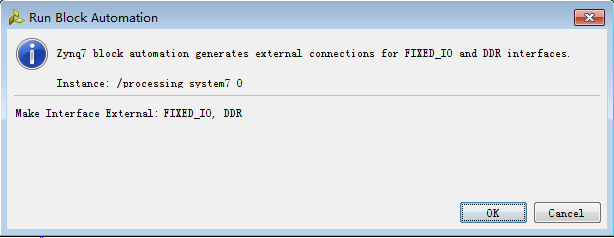
* + 1. 双击这个模块就可以打开定制模块的 Customization 窗口。



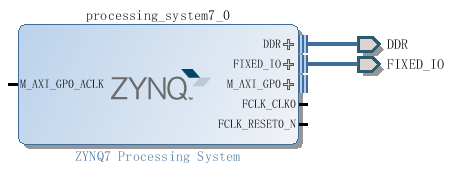
* 1. 配置处理模块，使得只有 UART 1 外设被启用。
     1. 点击Run Block Automation ，选择 /processing\_system7\_1



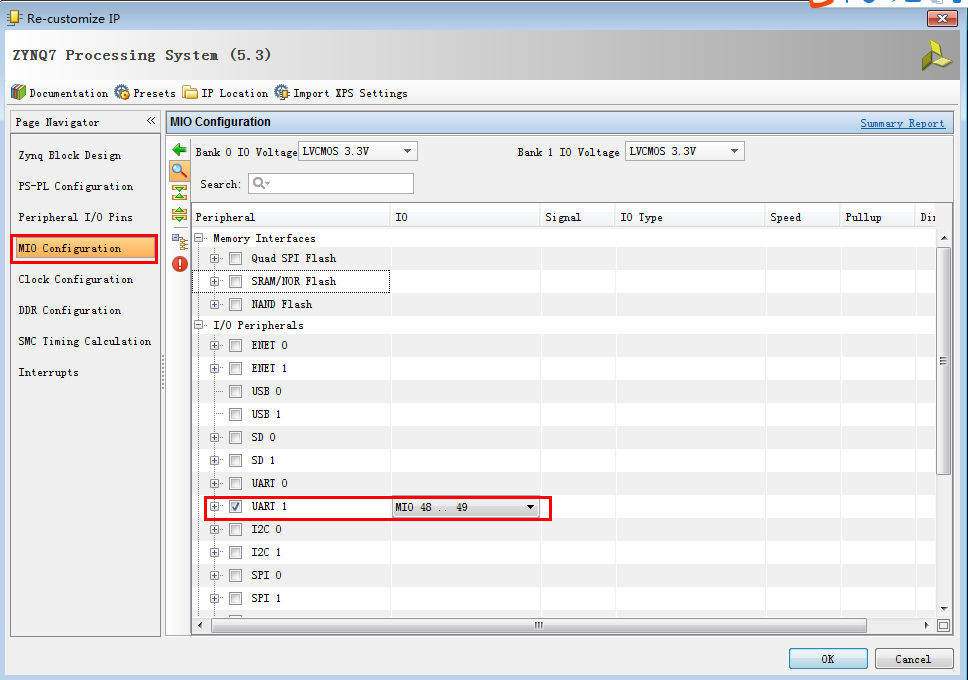
* + 1. 点击 OK。



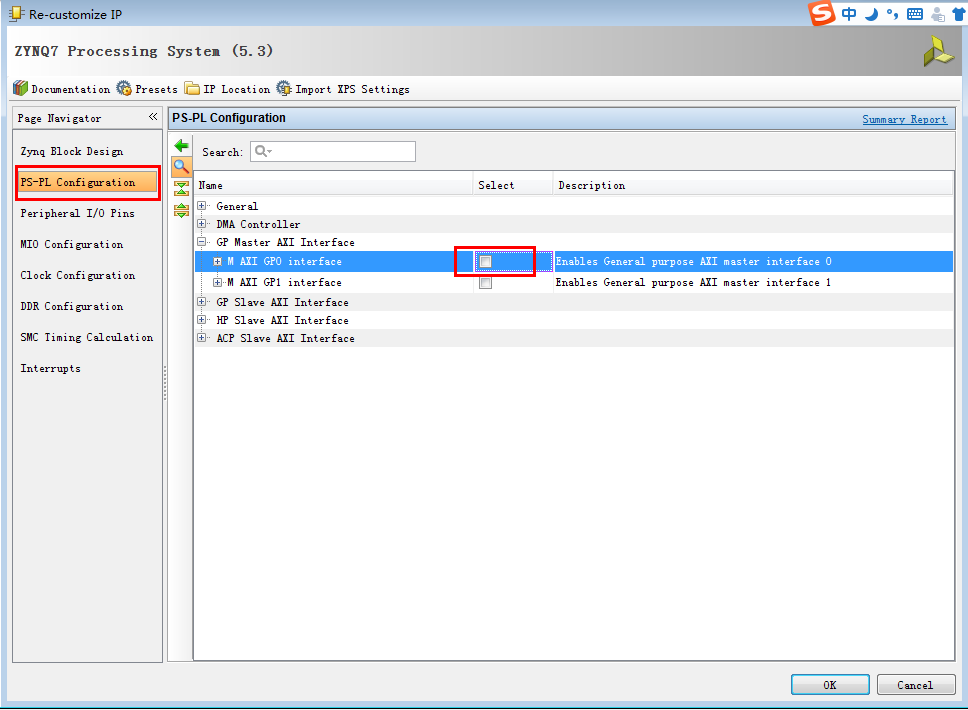
一旦模块自动化完成，可以看到DDR和Fixed IO 被自动添加了端口，一些额外的端口现在也可见了。



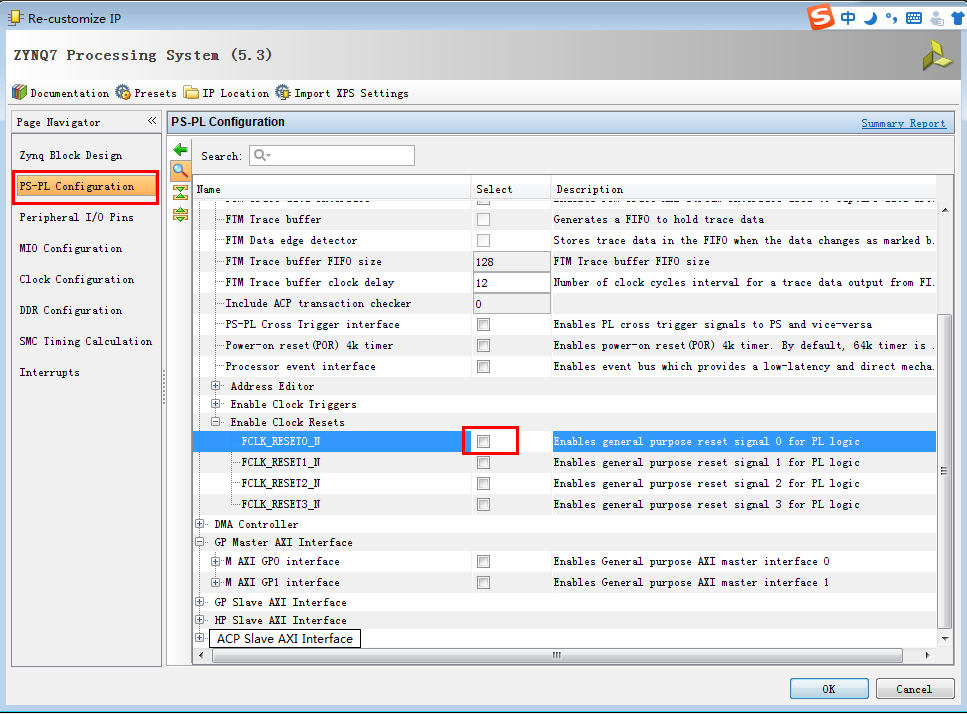
* + 1. 双击 Zynq 模块来打开Customization 窗口。
    2. 这里，设计者可以通过点击各种可配置模块（绿色，高亮）来改变系统配置。这次实验里只有UART被用到，所以其他的外设都不选。
    3. 选择左边的 MIO Configuration 标签， 打开配置表格。
    4. 将所有UART1以外的 I/O 外设都取消掉。



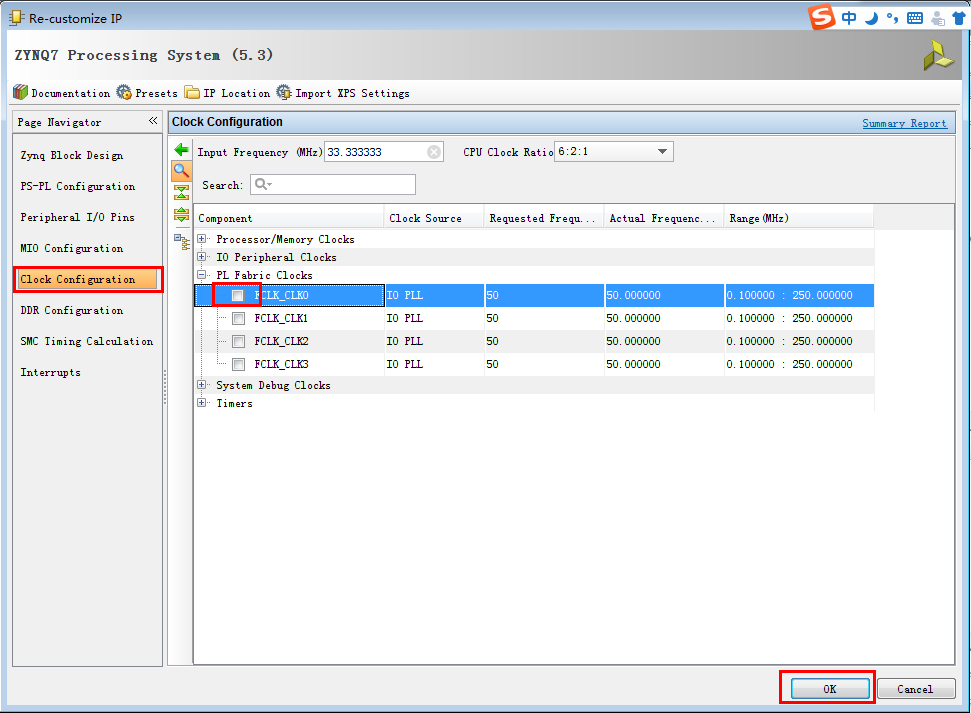
* + 1. 选择左边的 PS-PL Configuration 标签。 展开 GP Master AXI interface ，取消M AXI GP0 interface选项。



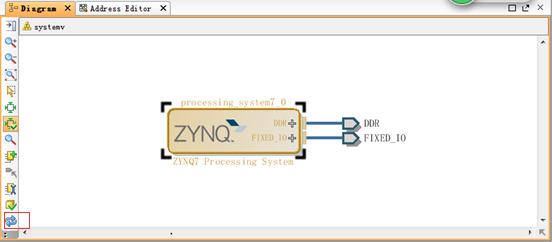
* + 1. 展开General > Enable Clock Resets，取消 FCLK\_RESET0\_N 选项。



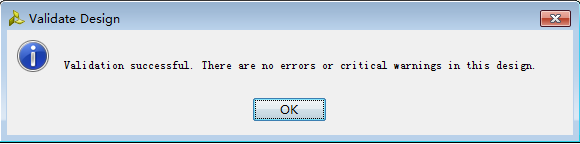
* + 1. 选择左边的 **Clock Configuration** 标签。展开**PL Fabric Clocks，**取消 **FCLK\_CLK0** 选项，点击OK。



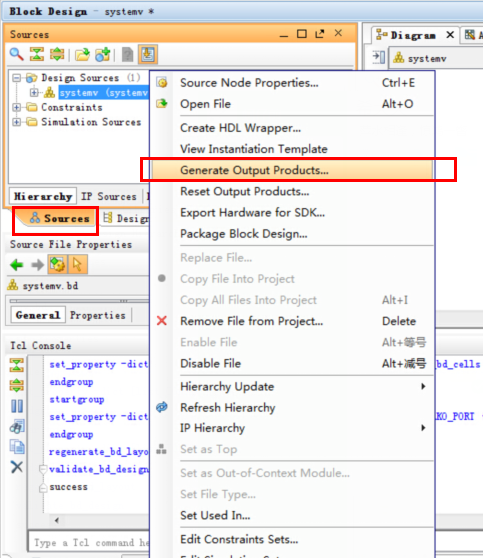
点击 (Regenerate Layout) 按钮 就可以看到下面的模块图。



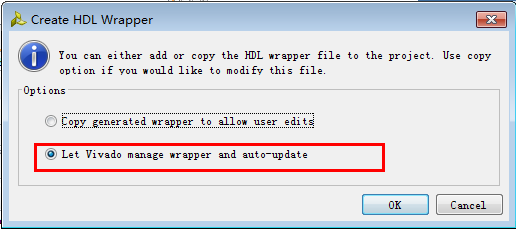
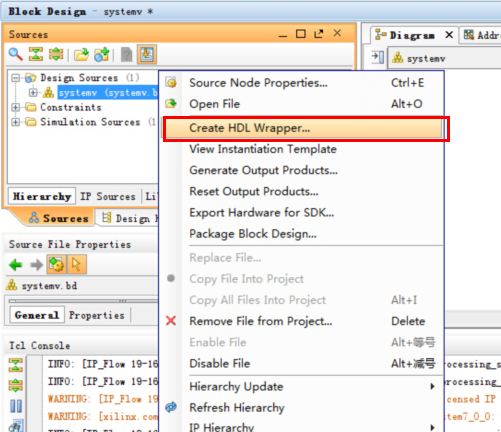
* + 1. 点击 (Validate Design) 按钮，确定设计没有出错。



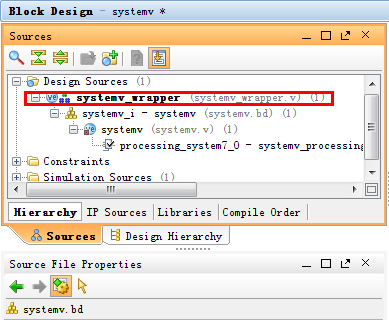
1. 生成顶层设计，导出到SDK Step 3
   1. 生成IP Integrator 输出, 顶层 Verilog HDL, 然后导出硬件以启动 SDK 。
      1. 在sources 面板中,右击 systemv.bd,选择Generate Output Products … 然后点击Generate 来生成次设计的实现，模拟与综合。



* + 1. 再次右击 systemv.bd, 选择 Create HDL Wrapper… 以生成顶层 Verilog HDL 模型。勾选Let Vivado manager wrapper and auto-update选项 , 然后点击 OK 。

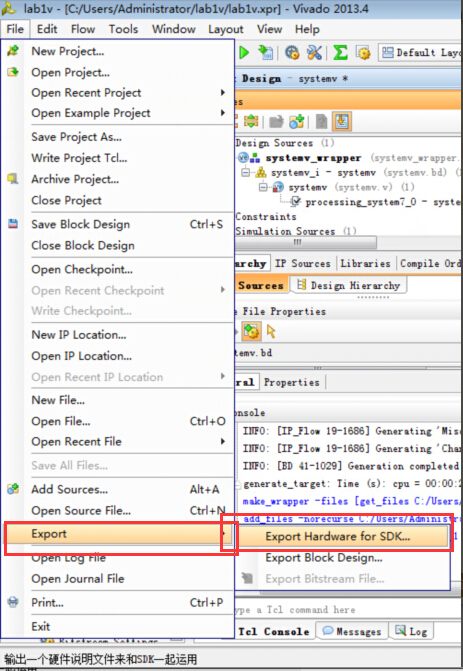


systemv\_wrapper.vhd file 将被创建并添加到工程中。

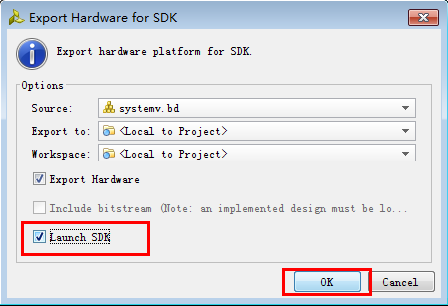


* + 1. 选择 File > Export > Export hardware for SDK…

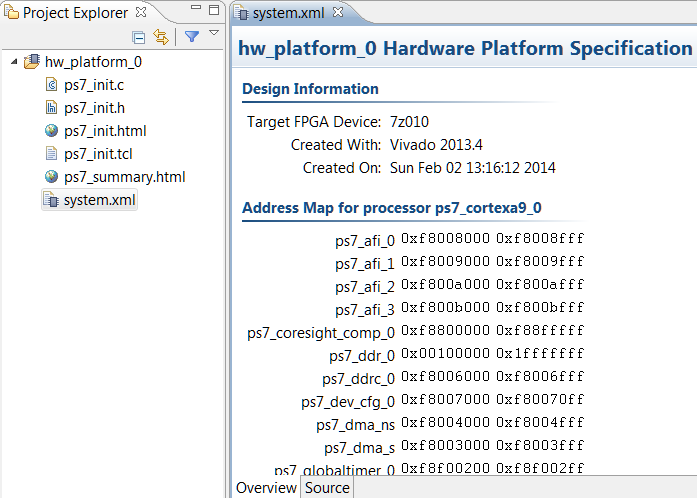
**导出设计到SDK前，你应将block design打开。 如果关掉了，点击 Flow Navigator pane面板里IP Integrator菜单下的Open Block Design选项来再次打开模块设计。**



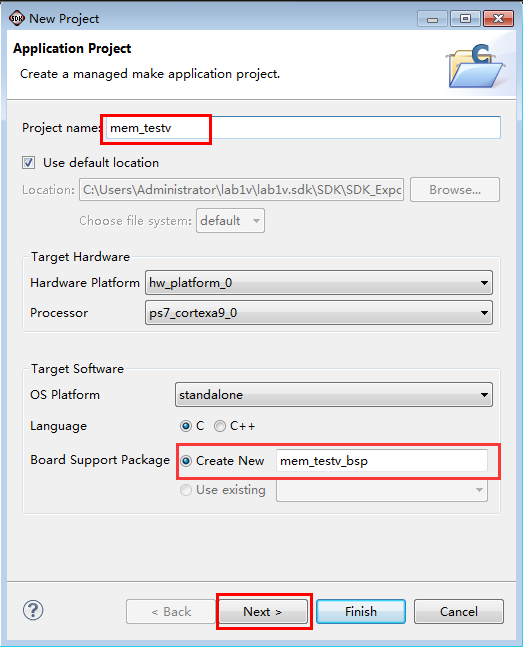
* + 1. 勾选 *Export Hardware和* Launch SDK ,点击 OK ，导出并启动 SDK。(如果有提示，保存设计)



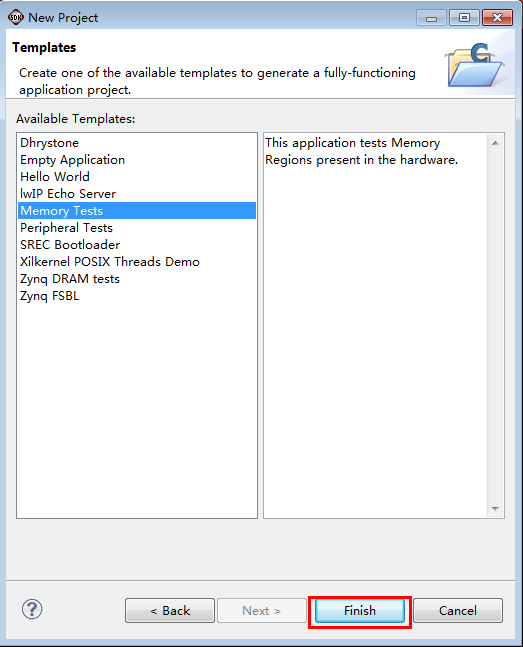
* + 1. SDK现在被打开了。 一个硬件平台工程已被创建， *hw\_platform\_0* 文件夹应该出现在 Project Explorer 面板中。



1. 在SDK里生成测试程序 Step 4
   1. 使用标准工程模板来生成内存测试程序。
      1. 在SDK中，选择 File > New > Application Project
      2. 将工程命名为 mem\_testv, 勾选 Create New selected 并保留默认名称 mem\_testv\_bsp ，点击Next。

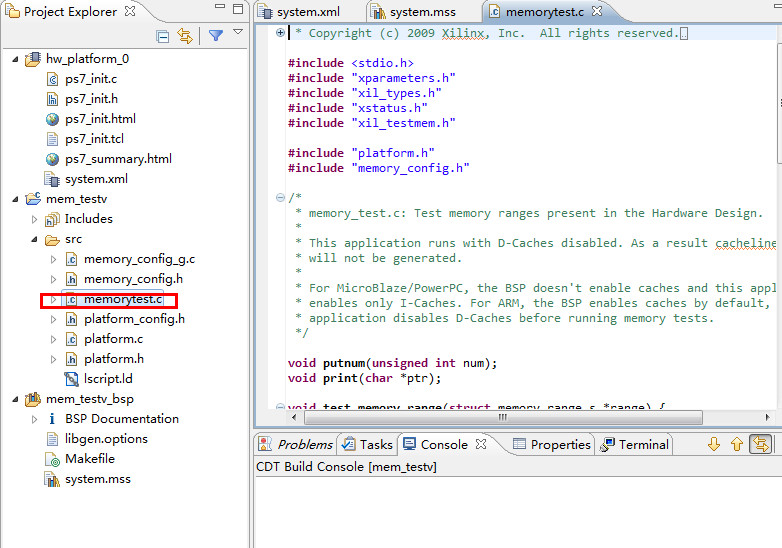


* + 1. 选择Memory Tests 模板，点击 Finish。

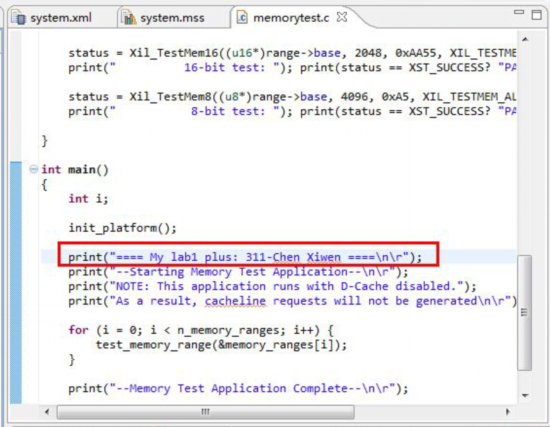


memv\_test 工程和板级支持工程 mem\_testv\_bsp 将被创建，并可以再SDK的 Project Explorer 窗口里看到，它们会被自动build 。 在 Console 面板里你可以看到这个过程。

* + 1. 展开 Project Explorer 视图下的文件夹，可以看到 有三个工程 - hw\_platform\_0, mem\_testv\_bsp, 以及 mem\_testv。mem\_testv 工程就是我们要测试设计功能的程序。hw\_platform 含有ps7\_init 函数，这个函数会将 PS 初始化为第一阶段bootloader的一部分，mem\_testv\_bsp 是板级支持包。.
    2. 打开 mem\_testv工程src文件架下的的 memorytest.c 文件，检查其中的内容。这个文件调用测试内存的函数。



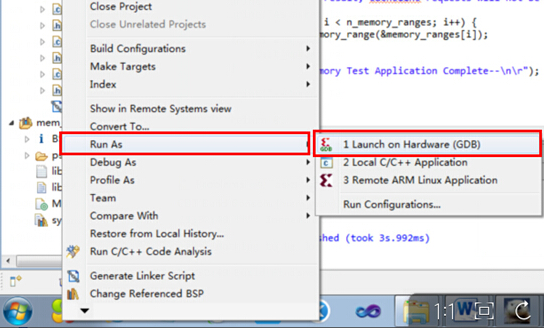
* + 1. 在memorytest.c文件的main函数中加入如下语句：



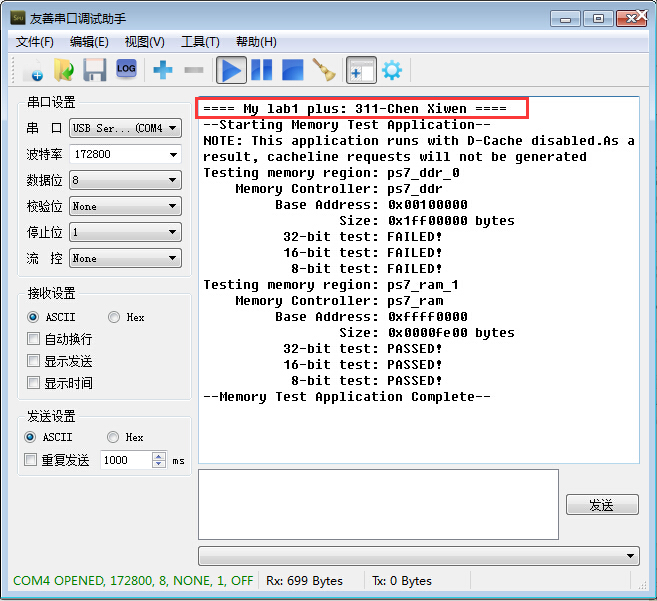
1. 在硬件上进行测试 Step 5
   1. 确保 JP7 被设置了，以选择USB 供电。将板子与一个micro-usb 线连接并上电。使用串口助手建立串行通讯。
      1. 确认 JP7被置为 USB 供电。
      2. 确定 micro-USB 线与JTAG PROG 连接器(紧邻供电连接器)连接。给板子上电。
      3. 打开串口助手，进行如下设置，点击“打开”。



* 1. 运行 mem\_testv 程序，测试功能。
     1. 在SDK里 选中 mem\_testv 工程，右击并选择Run As > Launch on Hardware (GDB) 来下载程序，执行 ps7\_init函数, 并运行 mem\_testv.elf。



* + 1. 在串口助手里就可以看到这样的信息。



* + 1. 分别都选择File > Exit，关闭SDK 和Vivado 。

结论

Vivado和 IP Integrator 能快速地生成基本的嵌入式处理器系统和应用程序。系统被定义后，Vivado就能导出硬件设计并调用SDK 。SDK提供了包括内存测试在内的多种应用程序，以便于进行软件开发。通过下载测试程序，将它在处理器上运行，并通过串口调试助手观察输出，你可以检验硬件的功能。