

PCIE 速度测试

黑金动力社区 2019-04-25

1 实验简介

PCIE (PCI Express) 采用了目前业内流行的点对点串行连接，比起 PCI 以及更早期的计算机总线的共享并行架构，每个设备都有自己的专用连接，不需要向整个总线请求带宽，而且可以把数据传输率提高到一个很高的频率，达到 PCI 所不能提供的高带宽。AX7103 开发板中的 FPGA xc7a100tfgg484-2 单通道通信速率可高达 5G bit 带宽，可配置成 X1、X2、X4 模式。该例程中通过利用 XILINX 的 XDMA IP 来实现 PCIE 的发送和接收速度测试。

2 实验原理

2.1 例程简介

Xilinx 的 xc7a100tfgg484-2 FPGA 集成了 1 个 PCIE 硬核，可配置成 X1、X2、x4 来实现不同速率的高速串行数据通信。在 AX7103 开发板上 PCIE 例程中，FPGA 端程序采用 XILINX 的 PCIE Core 进行设计，例中配置成 x4 进行 PCIE 通信，具体见 XILINX 官方 Vivado 文档的 PG195。PCIE 的相关基础知识可参考提供的《PCIE 概述》或查阅其它进行整体了解，对于已有基础的大虾来说，可忽略。

PCIE 通信例程由三部分组成：FPGA 端程序、PCIE 卡驱动、PCIE 上位机测试程序。

FPGA 端程序：负责建立与 PCIE 通信需具备的 FPGA 框架，PCIE 通信协议的构建；

PCIE 卡驱动：负责上位机测试程序与 PCIE 卡的数据交换；

PCIE 上位机测试程序：PCIE 测速（如 linux 下开发，请参考实验教程《ALINX 黑金 PCIE 板

卡

Linux 使用教程》）。

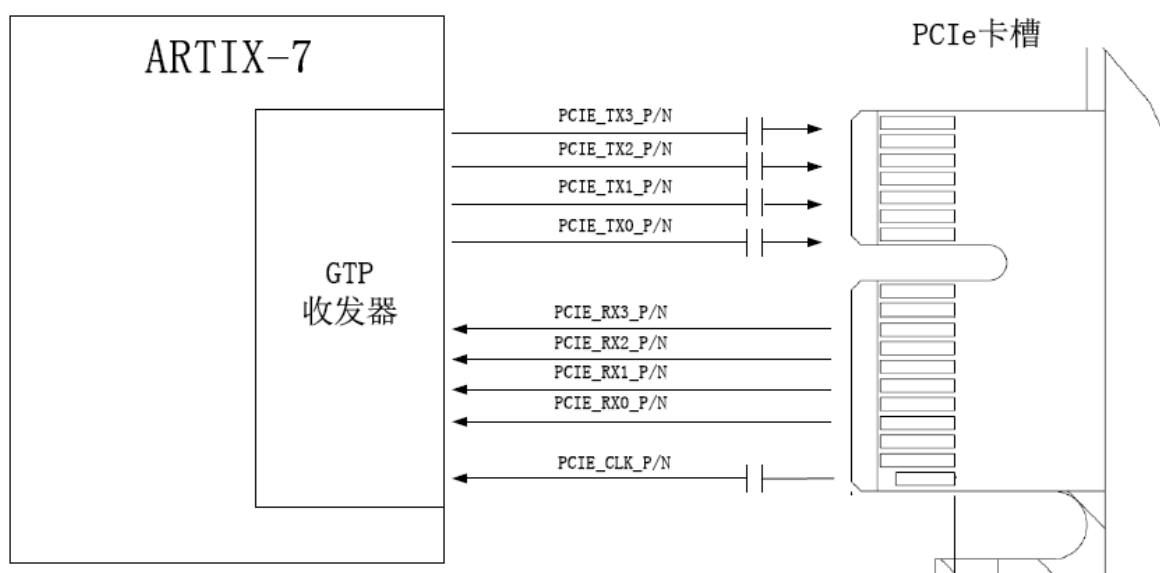
在进行 PCIE 测速例程之前，确保计算机为 WIN7（64 位）或 WIN10（64 位）系统。

2.2 硬件描述

AX7103 扩展板上提供一个工业级高速数据传输 PCIe x4 接口，PCIe 卡的外形尺寸符合标准 PCIe 卡电气规范要求，可直接在普通 PC 的 x4 PCIe 插槽上使用。

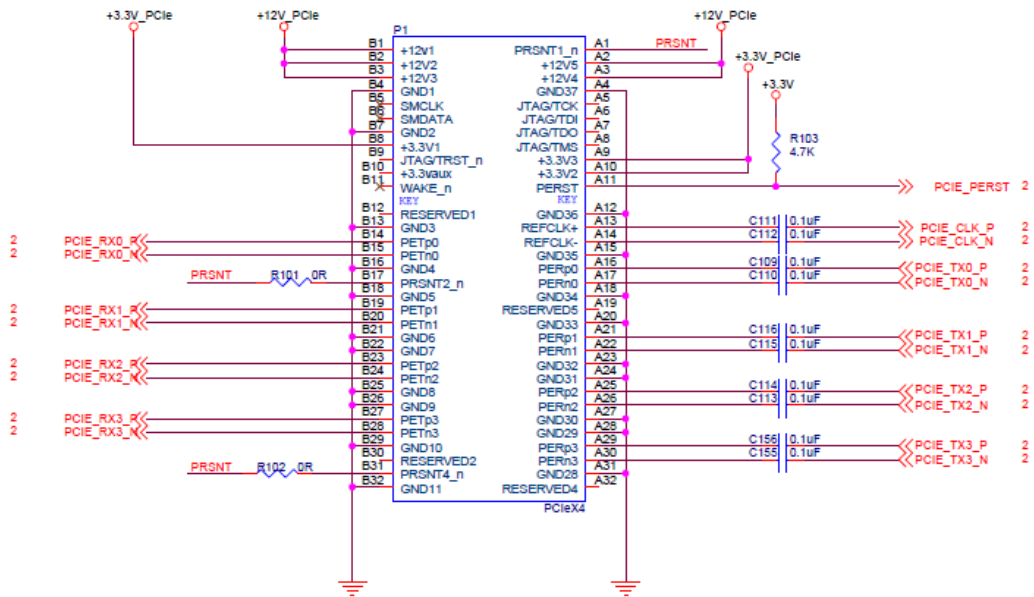
PCIe 接口的收发信号直接跟 FPGA 的 GTP 收发器相连接，四通道的 TX 信号和 RX 信号都是以差分信号方式连接到 FPGA，单通道通信速率可高达 5G bit 带宽。PCIe 的参考时钟由 PC 的 PCIe 插槽提供给开发板，参考时钟频率为 100Mhz。

开发板的 PCIe 接口的设计示意图如下图所示,其中 TX 发送信号和参考时钟 CLK 信号用 AC 耦合模式连接。



在电路设计中硬件电路部分：

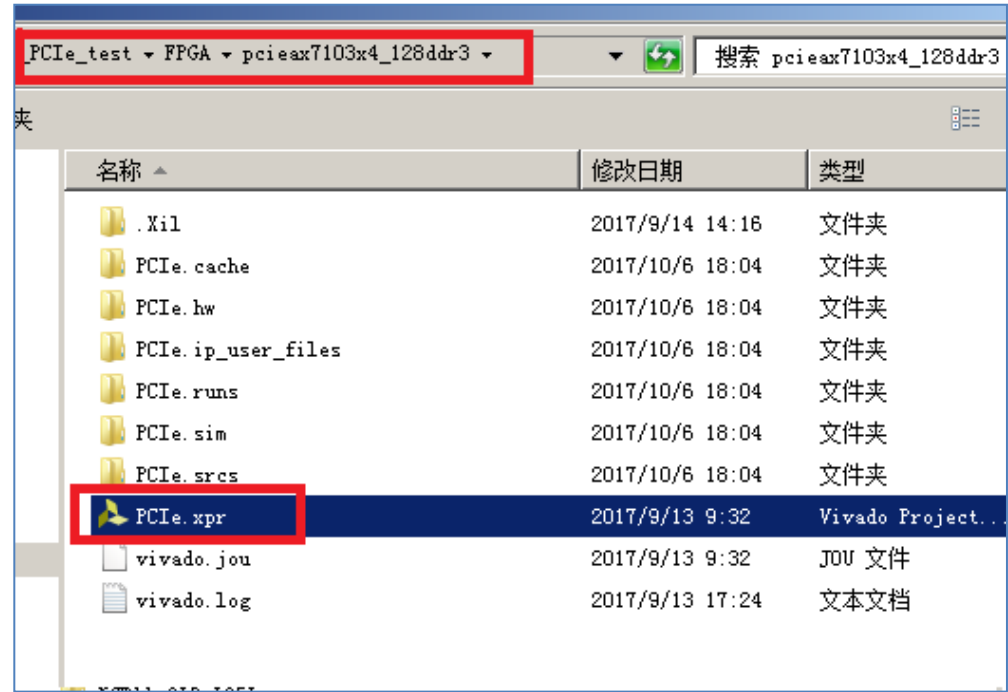
PCIE X4 SLOT



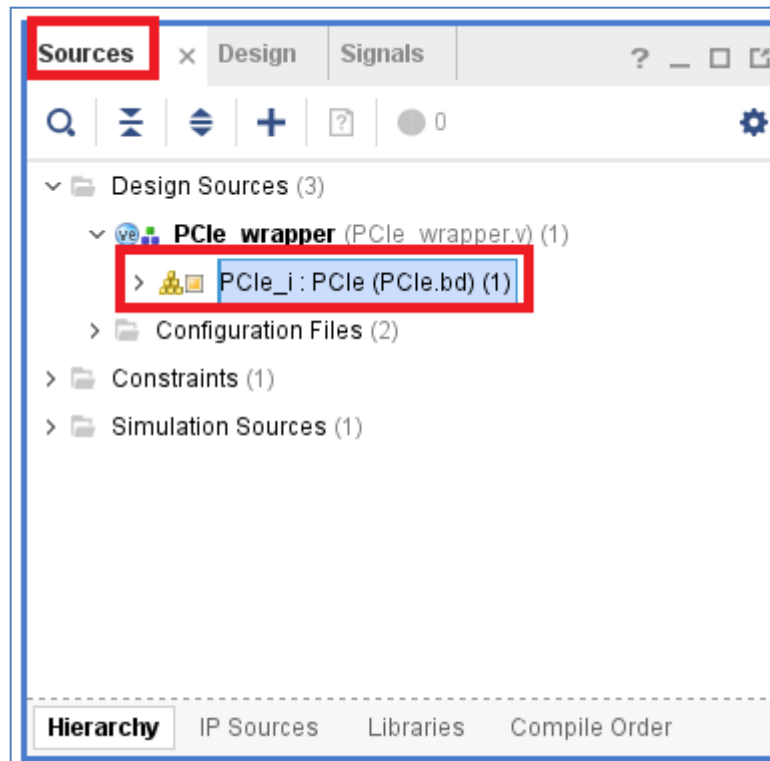
3 程序设计

3.1 FPGA 程序

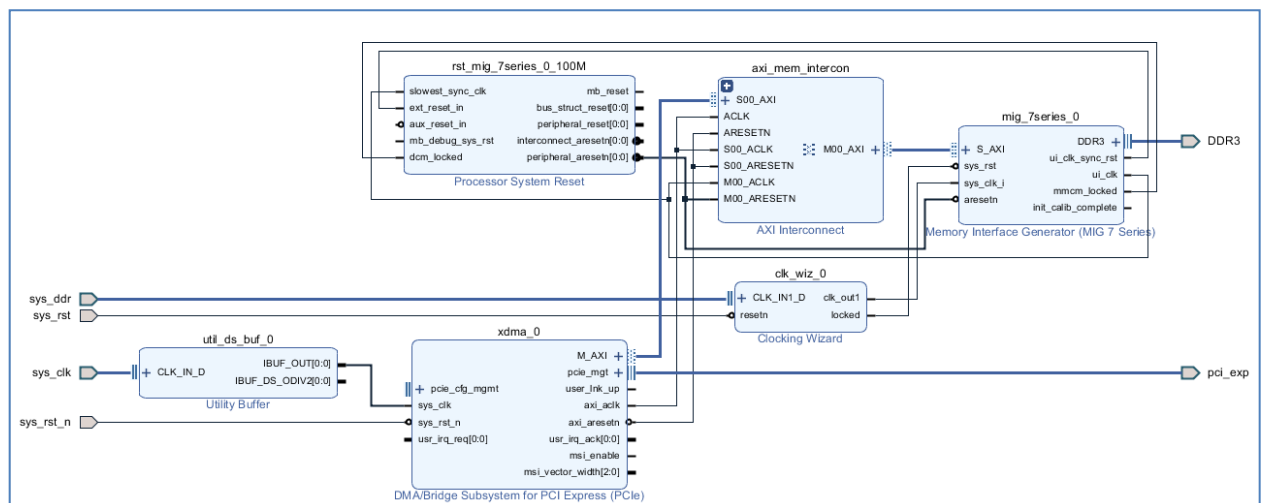
打开 Vivado 的 FPGA 工程，FPGA 程序位于如下图位置



设计中采用 XILINX 的 Block Design，打开 Pcie.bd。



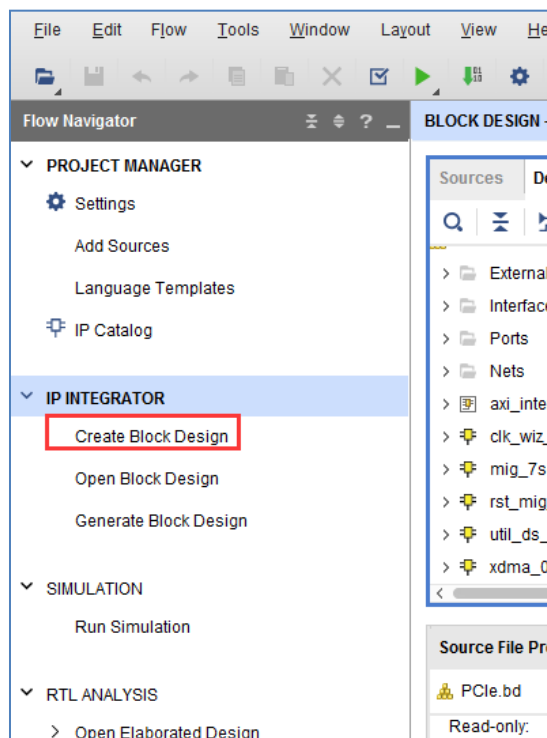
可看到 PCIe 的 FPGA 代码，这里我们采样的是原理框图的设计方法，PCIE 只要调用 xilinx 提供的 IP，设计就这么简单！！



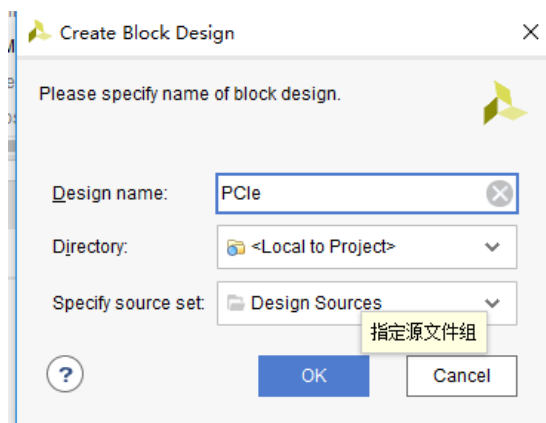
程序模块说明：PCIe 通信程序由 util_ds_buf_0、clk_siz_0、rst_mig_7series_0_100M、axi_interconnect、mig_7series_0 及 XDMA_0 组成。util_ds_buf_0 是对外部的 PCIe 差分输入时钟进行 buffer；clk_siz_0 负责为 mig_7series_0（DDR3）模块提供时钟及复位信号；rst_mig_7series_0_100M 是复位模块为 axi_interconnect_0 和 mig_7series_0 提供参考信号；axi_interconnect_0 是 AXI 的 Master 和 Slave 接口设备互联的协议模块；XDMA_0 模块是 PCIe 通信模块，内部具备 DMA 功能，只在 Vivado2016 以上版本才具备。

下面对 PCIe.bd 的创建及各个模块配置进行介绍：

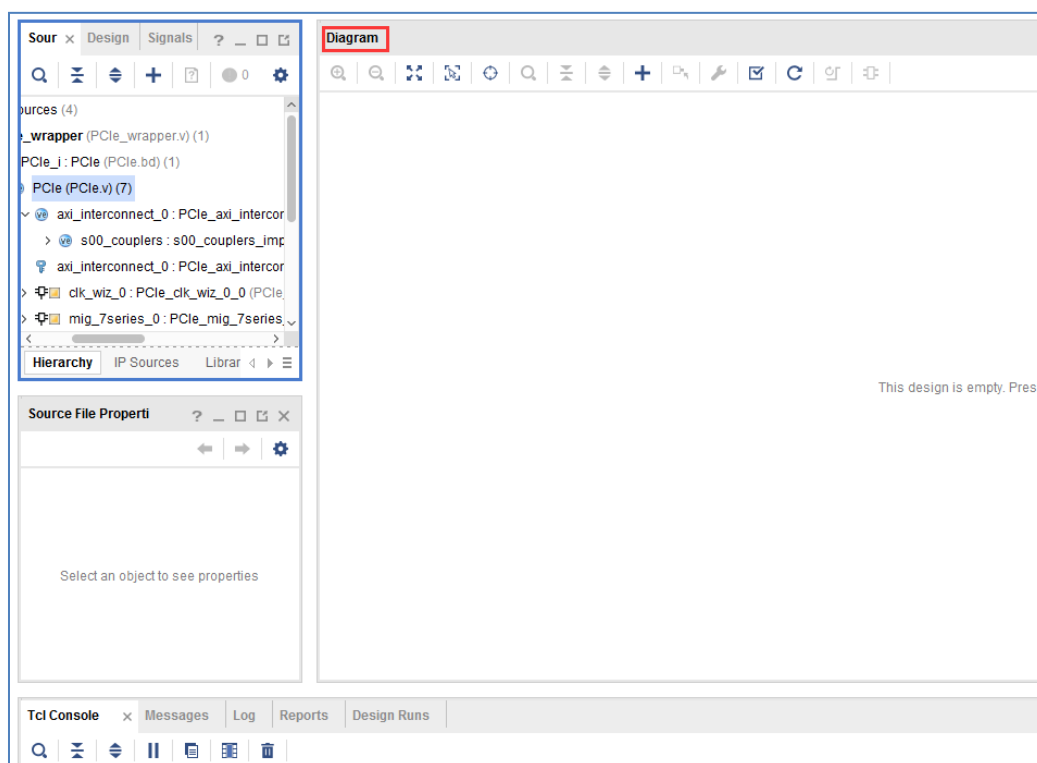
1. 在如下图中选择 Create Block Design, 这里使用原理框图的设计。



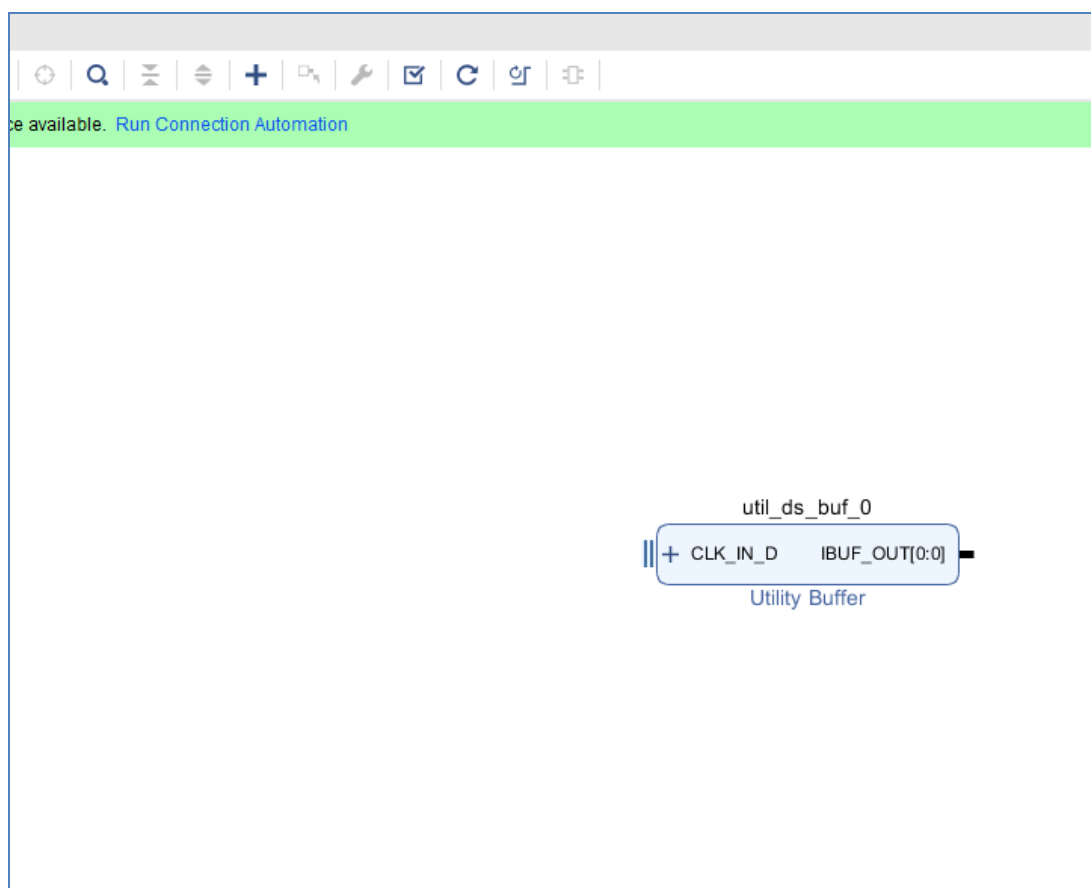
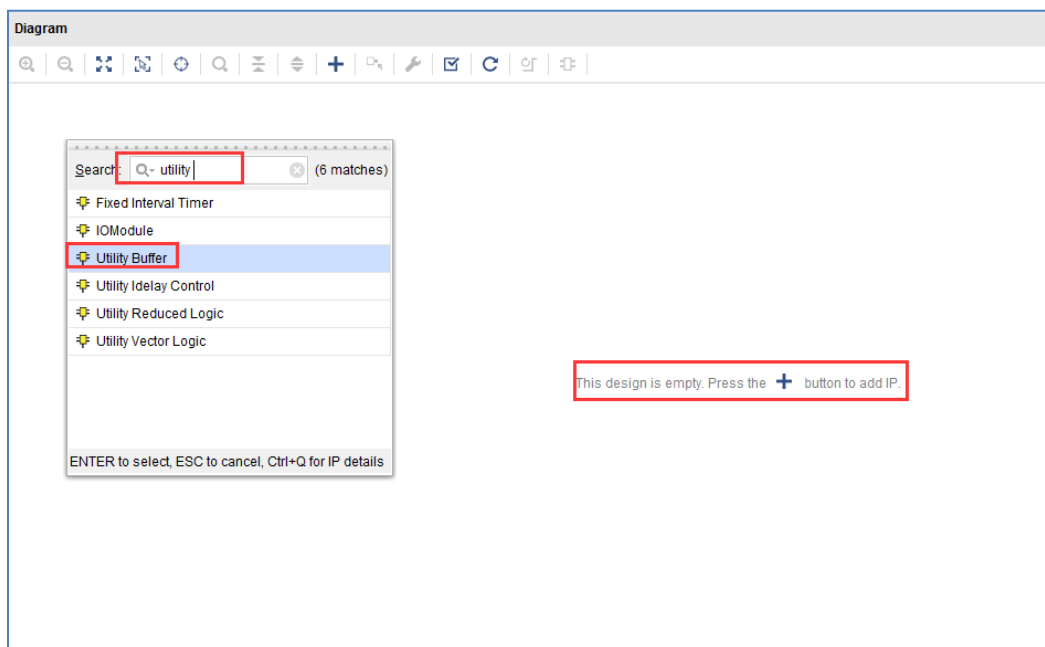
在弹出对话框选择取名 PCIe，如下图：



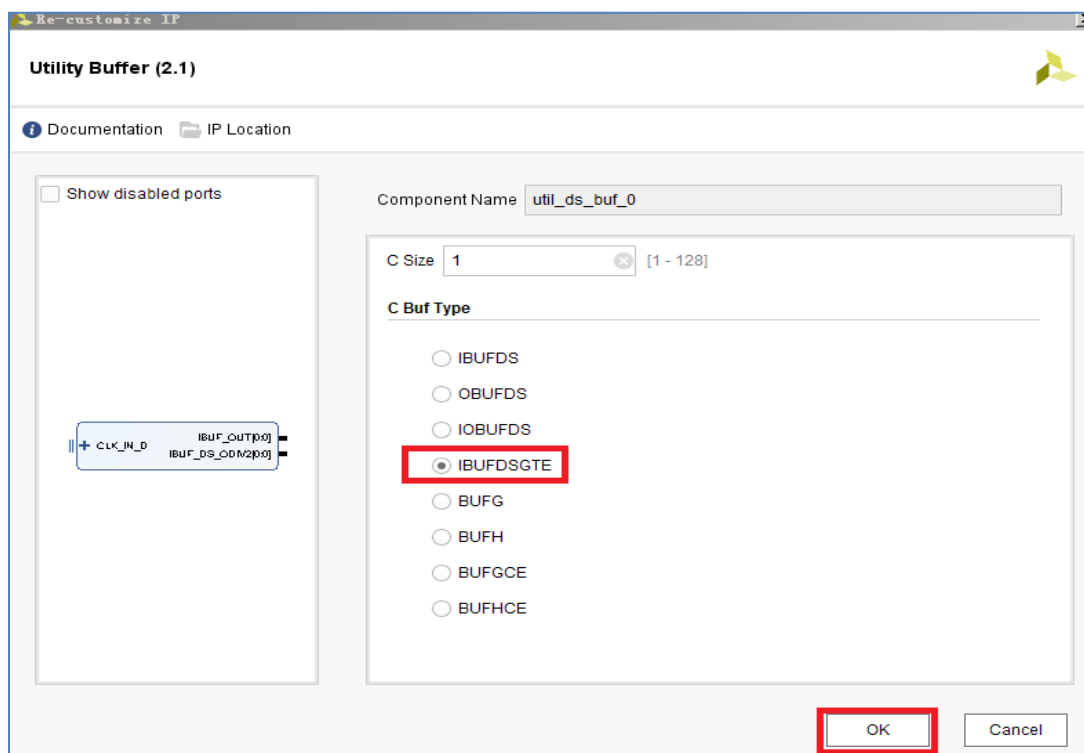
创建完成后可看到如下的空白窗体：



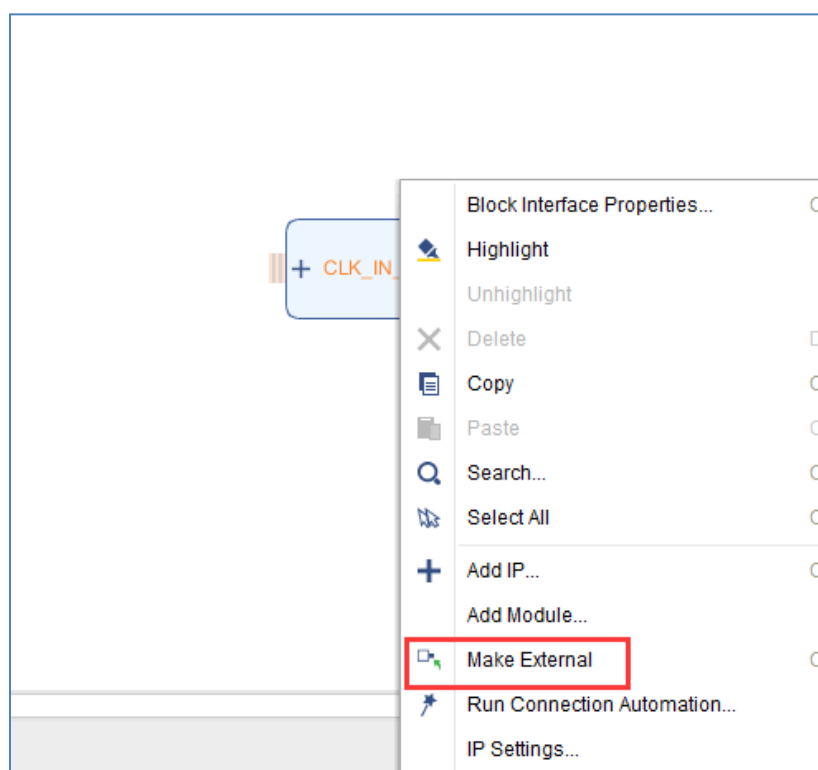
2. 创建第一个 utility Buffer,作用是连接 PCIe 的外部参考时钟的，单击 “+” 或右键选择 “add IP”，在弹出的对话框输入 utility 即可找到并双击 utility Buffer，如下图中所示：

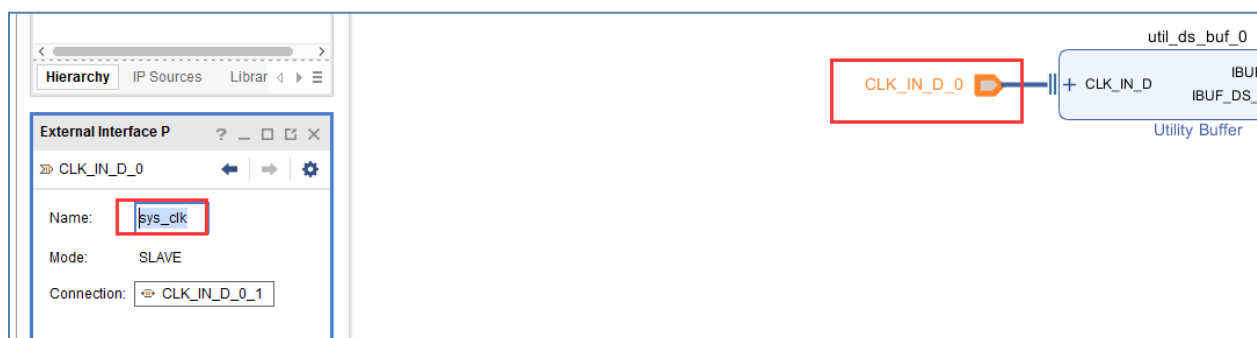


其配置如下，双击窗体中的 utility Buffer 打开后，因为这里是 GTP 收发器的差分时钟,需要选择 “IBUFDSGTE”，按 OK。



接下来给模块的 CLK_IN_D 管脚分配外部时钟接口，鼠标指向接口并右键选择 Make External 即可完成，管脚改名为 sys_clk 如下图中所示：





3. Clocking Wizard 模块的创建，按照上面讲述的方法添加模块，作用是接收外部 200MHz 的差分时钟，为 MIG 控制器的提供输入时钟，其配置如下：

The screenshot shows the 'Clocking Wizard' configuration window for component 'clk_wiz_0'. The 'Clocking Options' tab is selected. The 'Clock Monitor' section has 'Enable Clock Monitoring' unchecked. The 'Primitive' is set to 'MMCM'. The 'Clocking Features' section has 'Frequency Synthesis' and 'Phase Alignment' checked. The 'Jitter Optimization' section has 'Balanced' selected. The 'Dynamic Reconfig Interface Options' section has 'AXI4Lite' selected. The 'Input Clock Information' table shows two input clocks: Primary (clk_in1, 200.000 MHz) and Secondary (clk_in2, 100.000 MHz).

Input Clock	Port Name	Input Frequency(MHz)	Jitter Options	Input Jitter	Source
Primary	clk_in1	200.000	UI	0.010	Differential clock capable pin
Secondary	clk_in2	100.000		0.010	Single ended clock capabl...

输出时钟为单端 200Mhz。

Component Name

Clocking Options

Output Clock

MMCM Settings

Summary

		Requested	Actual	Requested	Actual	Requested	Actual	
<input checked="" type="checkbox"/>	clk_out1	clk_out1	200.000	200.000	0.000	0.000	50.000	50.0 BUFG
<input type="checkbox"/>	clk_out2	clk_out2	100.000	N/A	0.000	N/A	50.000	N/A BUFG
<input type="checkbox"/>	clk_out3	clk_out3	100.000	N/A	0.000	N/A	50.000	N/A BUFG
<input type="checkbox"/>	clk_out4	clk_out4	100.000	N/A	0.000	N/A	50.000	N/A BUFG
<input type="checkbox"/>	clk_out5	clk_out5	100.000	N/A	0.000	N/A	50.000	N/A BUFG
<input type="checkbox"/>	clk_out6	clk_out6	100.000	N/A	0.000	N/A	50.000	N/A BUFG
<input type="checkbox"/>	clk_out7	clk_out7	100.000	N/A	0.000	N/A	50.000	N/A BUFG

☐ USE CLOCK SEQUENCING

Clocking Feedback

Output Clock	Sequence Number
clk_out1	1
clk_out2	1
clk_out3	1
clk_out4	1
clk_out5	1
clk_out6	1
clk_out7	1

Source

Signaling

☒ Automatic Control On-Chip
☐ Automatic Control Off-Chip
☐ User-Controlled On-Chip
☐ User-Controlled Off-Chip

☒ Single-ended
☐ Differential

Enable Optional Inputs / Outputs for MMCM/PLL

Reset Type

☒ reset ☐ power_down ☐ input_clk_stopped

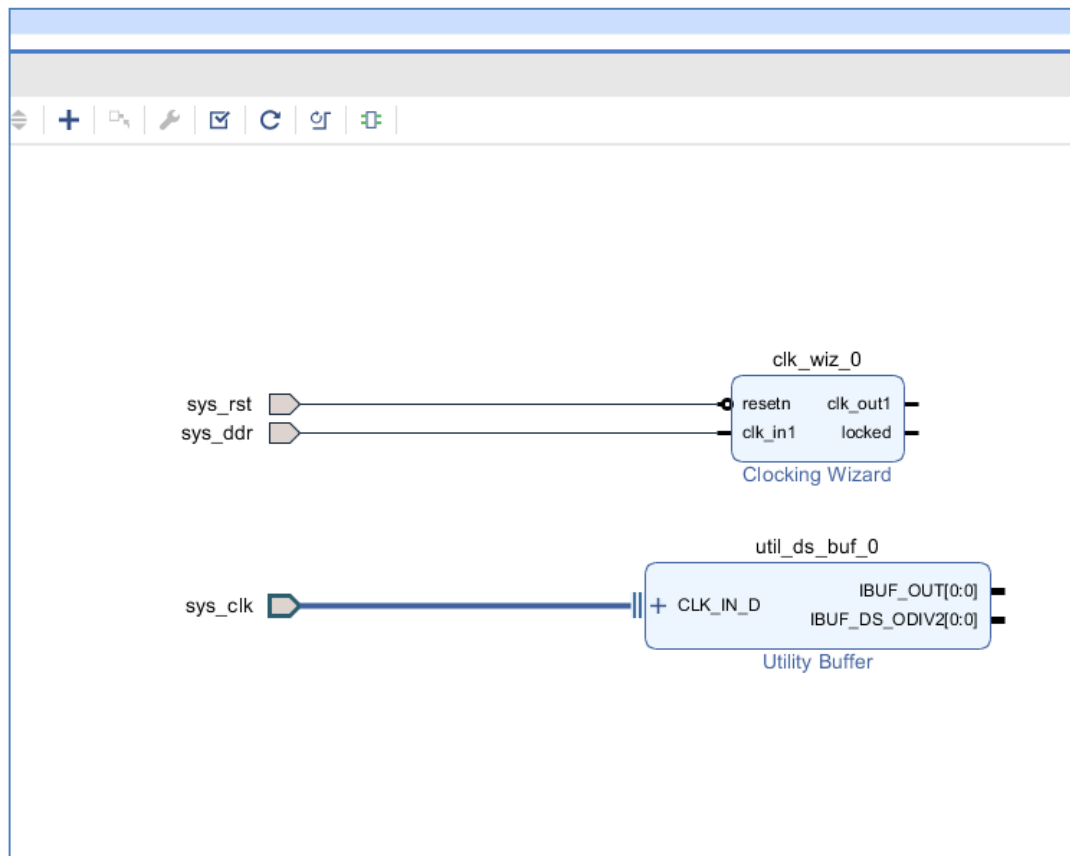
☐ Active High ☒ Active Low

☒ locked ☐ clkfbstopped

OK

Cancel

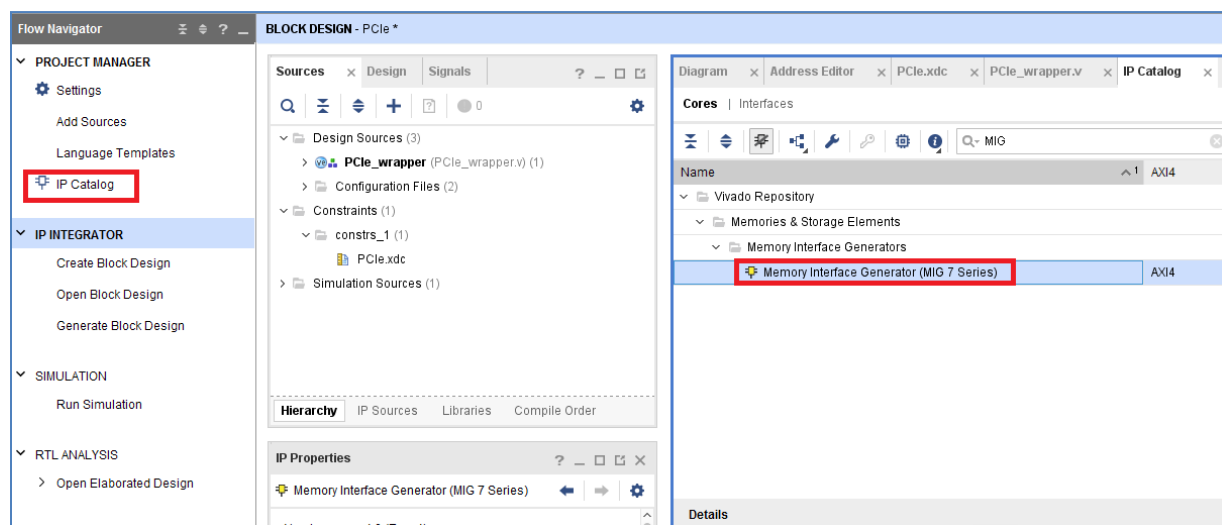
完成管脚修改后如下图：



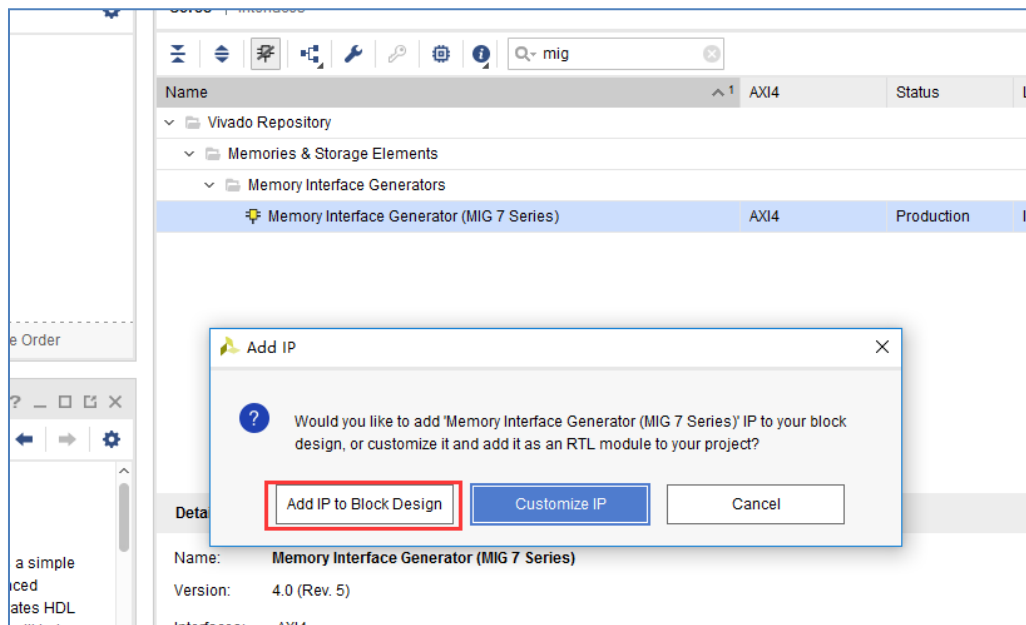
4. MIG 控制器的创建，这个模块的作用是控制 DDR3 工作的，配置如下：

mig_7series_0 配置如下：

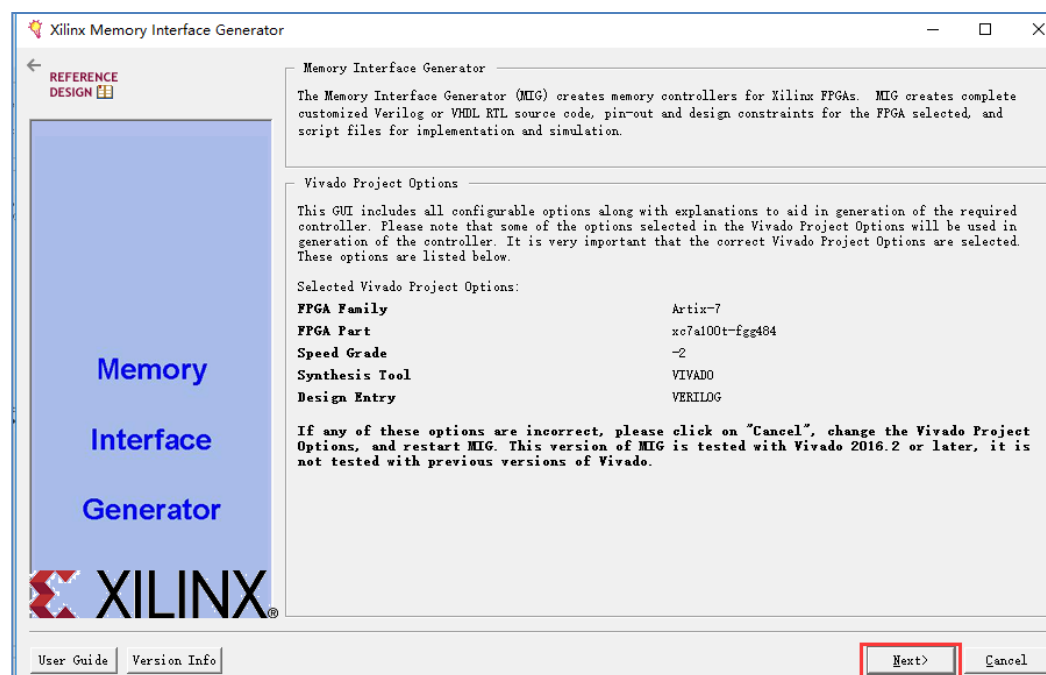
1) 添加双击 Memory Interface Genrator 模块



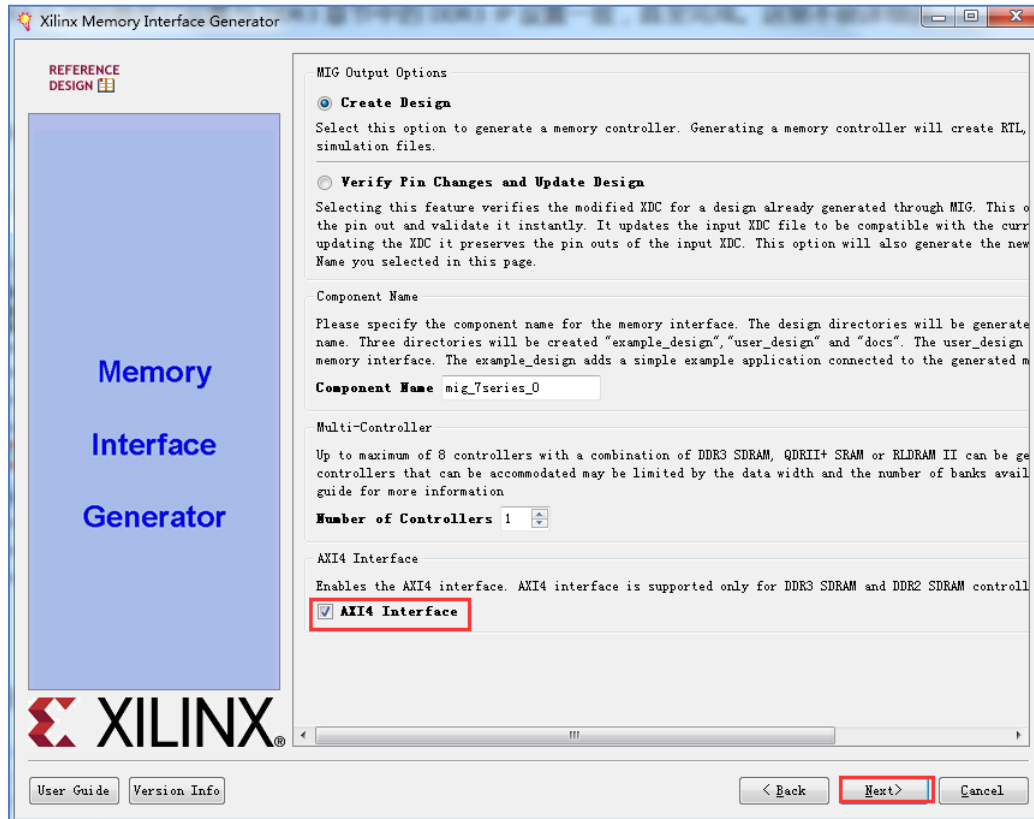
2) 在弹出的对话框中选择如下：



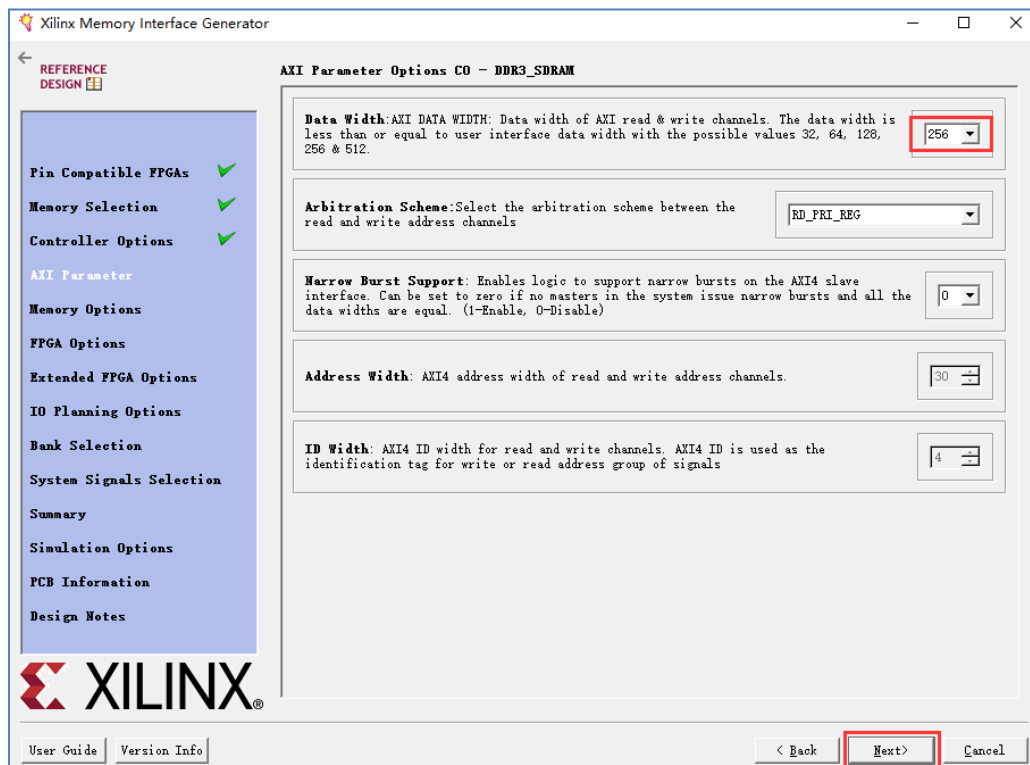
3) 在如下界面下单击 Next



4) 由于要用到 AXI 总线，选择 AXI4 Interface 接口，单击 Next

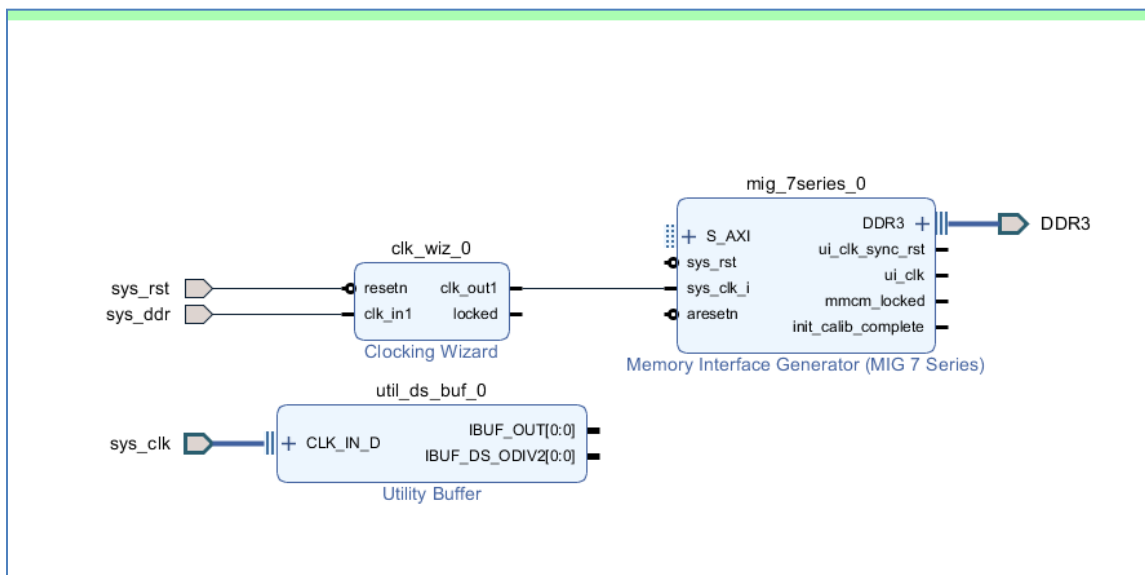


5) 在配置 AXI4 DDR3 IP 控制器过程中会出现如下新的界面，设置如图，然后单击 Next：



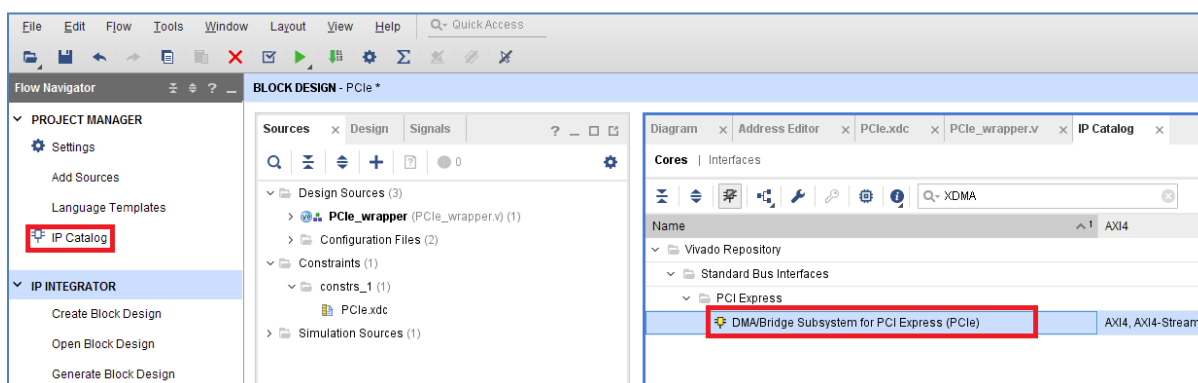
6) DDR3 IP 控制器其它设置与 DDR3 读写测试章节中的 DDR3 IP 设置类似，直至完成。这里不做详细讲述。

进行简单的连接及管脚修改完成后如下：

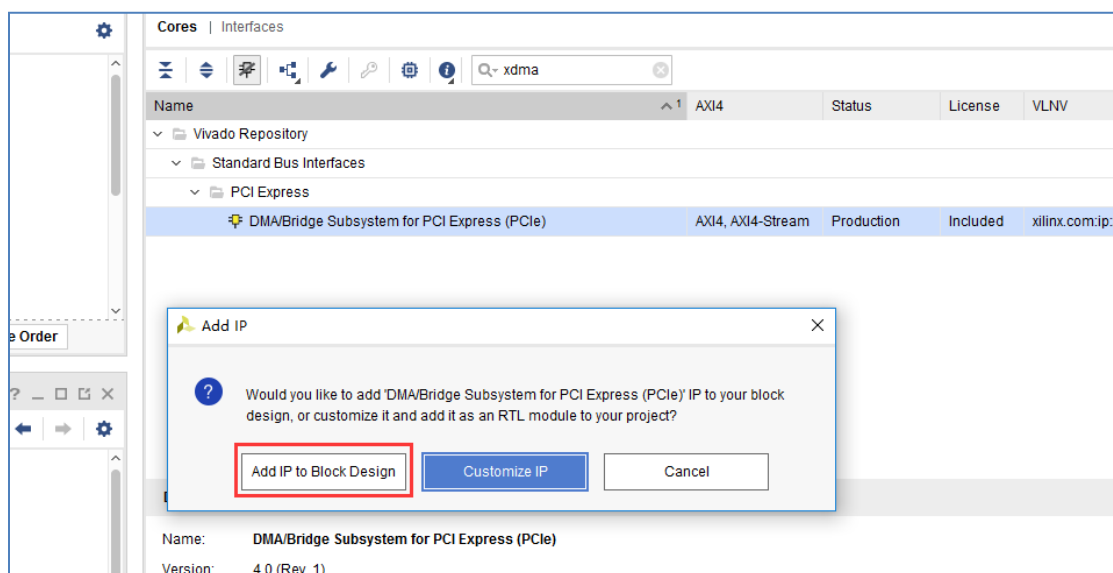


5. XDMA_0 模块配置如下：

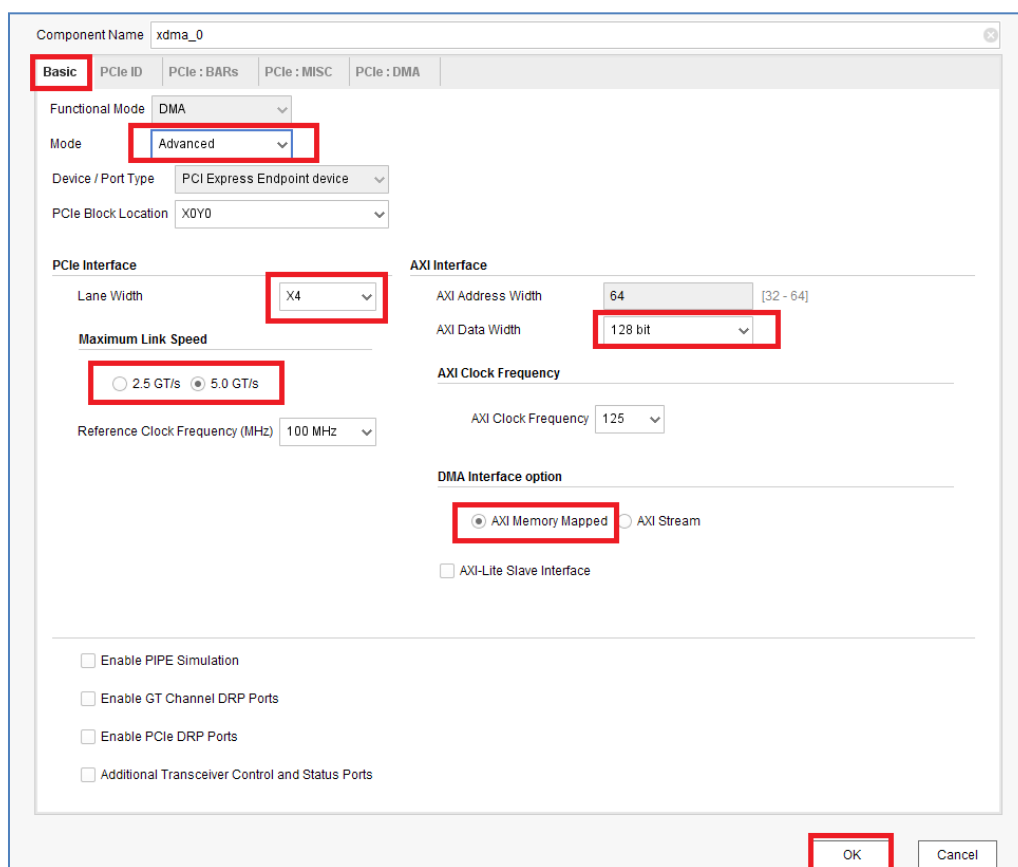
1) 在 IP Catalog 中找到如下图中的 PCIe，并双击。



2) 在弹出的对话框中选择 “Add IP to Block Design”：



3) 按下图进行 Basic 栏设置，这里选择主要是 Mode、PCIe Lane、Link Speed、DMA 接口方式选择等，按图中设置即可。



4) PCIe ID 栏按下图红色框中设置，其它默认：

Component Name: xdma_0

Basic | **PCIe ID** | PCIe : BARs | PCIe : MISC | PCIe : DMA

ID Initial Values

Vendor ID	10EE	✕
Device ID	7024	✕
Revision ID	00	✕
Subsystem Vendor ID	10EE	✕
Subsystem ID	0007	✕

Class Code Lookup Assistant

☐ Use Class Code Lookup Assistant

Base Class Menu	Simple communication controllers	▼
Base Class Value	05	✕ Range: 00..FF
Sub Class Interface Menu	Generic XT compatible serial controller	▼
Sub Class Value	80	✕ Range: 00..FF
Interface Value	00	✕ Range: 00..FF
Class Code	058000	Range: 000000..FFFFFF

5) PCIe:BARs、PCIe:MISC 中保持默认设置。

Component Name: xdma_0

Basic | PCIe ID | **PCIe : BARs** | PCIe : MISC | PCIe : DMA

☐ PCIe to AXI Lite Master Interface

☐ 64bit Enable ☐ Prefetchable

Size: 4 Scale: Kilobytes

Value: FFFF000

PCIe to AXI Translation: 0x0000000000000000

☒ PCIe to DMA Interface

☐ 64bit Enable ☐ Prefetchable

☐ PCIe to DMA Bypass Interface

☐ 64bit Enable ☐ Prefetchable

Size: 1 Scale: Megabytes

Value: FFF00000

PCIe to AXI Translation: 0x0000000000000000

Component Name

Basic	PCIe ID	PCIe : BARs	PCIe : MISC	PCIe : DMA
-------	---------	-------------	-------------	------------

User Interrupts
Number of User Interrupts Request (1-16)

Legacy Interrupt Settings
Legacy Interrupt Settings

MSI Capabilities
☒ Enable MSI Capability Structure
Multiple Message Capability

MSI-X Capabilities
☐ Enable MSI-X Capability Structure

Miscellaneous
☐ Finite Completion Credits
☒ Extended Tag Field
☒ Configuration Management Interface

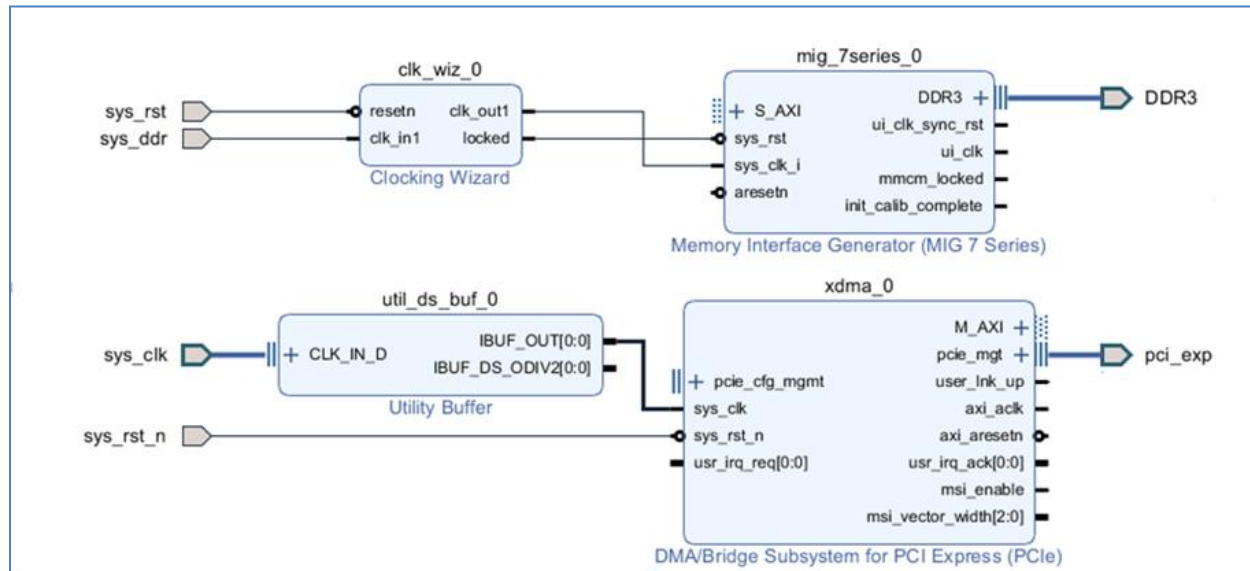
Link Status Register
Selects whether the device reference clock is provided by the connector (Synchronous) or generated via an onboard PLL(Asynchronous)
☒ Enable Slot Clock Configuration

6) PCIe:DMA 主要设置 DMA 的参数：H2C 通道数、C2H 通道数、ID 设置，采用下图设置即可，然后单击 OK 即可完成设置。

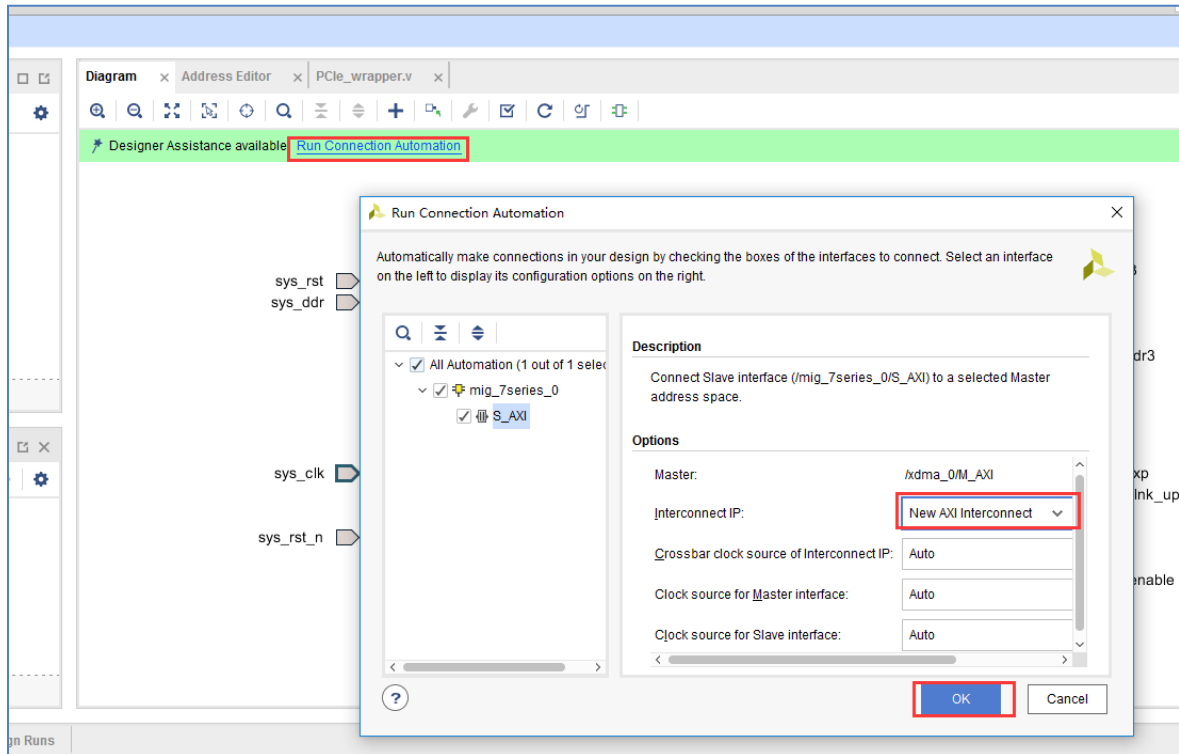
Component Name:

Basic	PCIe ID	PCIe : BARs	PCIe : MISC	PCIe : DMA
Number of DMA Read Channel (H2C)				
<input type="text" value="2"/>				
Number of DMA Write Channel (C2H)				
<input type="text" value="2"/>				
Number of Request IDs for Read channel (2,4,8,16,32,64)				
<input type="text" value="32"/> [2 - 64]				
Number of Request IDs for Write channel (2,4,8,16,32)				
<input type="text" value="16"/> [2 - 32]				
Descriptor Bypass for Read (H2C)				
<input type="text" value="0000"/>				
Descriptor Bypass for Write (C2H)				
<input type="text" value="0000"/>				
AXI ID Width				
<input type="text" value="4"/>				
<input type="checkbox"/> DMA Status Ports				

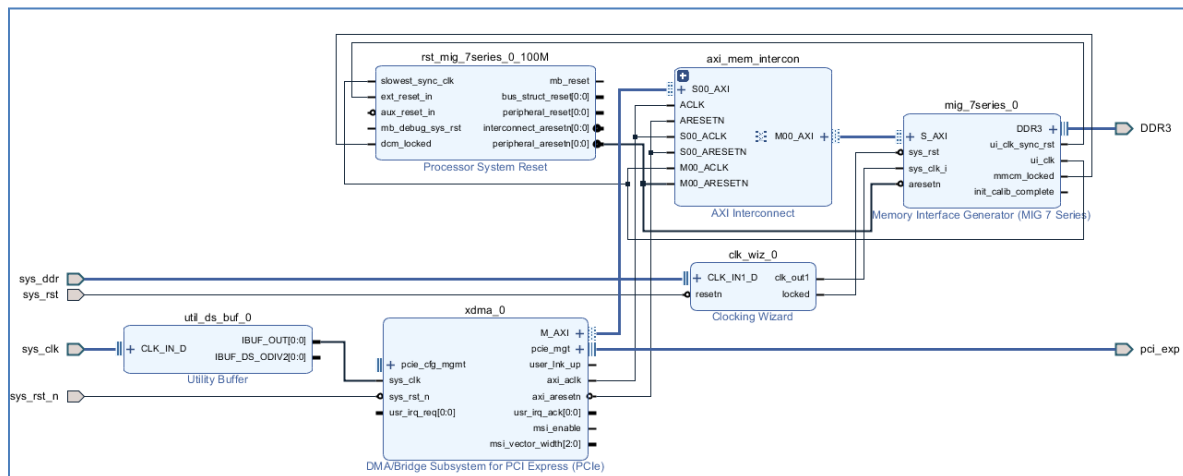
完成后并进行简单的连线后如下图所示：



6. 单击下图中的 Run Connection Automation,并进行如下设置，再单击 OK 后会自动进行连线，



完成后如下所示：



rst_mig_7series_0_100M 配置采用默认即可。

7. axi_interconnect 配置如下，模块的具体内容见 XILINX pg059-axi-interconnect.pdf 文档；

Re-customize IP

AXI Interconnect (2.1)

Documentation IP Location

Component Name: axi_interconnect_0

Top Level Settings Slave Interfaces Advanced Options

Number of Slave Interfaces: 1

Number of Master Interfaces: 1

Interconnect Optimization Strategy: Custom

AXI Interconnect includes IP Integrator automatic converter insertion and configuration.

When the endpoint IPs attached to the interfaces of the AXI Interconnect differ in width, clock or protocol, a converter IP will automatically be added inside the interconnect. If a converter IP is inserted, IP integrator's parameter propagation automatically configures the converter to match the design.

To see which conversion IPs have been inserted, use the IP integrator 'expand hierarchy' buttons to explore inside the AXI Interconnect hierarchy.

NOTE: Addressing information for AXI Interconnect is specified in the IP Integrator address editor.

☒ Enable Advanced Configuration Options

OK Cancel

Re-customize IP

AXI Interconnect (2.1)

Documentation IP Location

Component Name: axi_interconnect_0

Top Level Settings Slave Interfaces **Advanced Options**

Clock Domain Crossing MTBF Options

Synchronization Stages: 2

Interconnect Crossbar Options

Data Width of the AXI Crossbar: 256

Interconnect Debug Options

☐ Enable Protocol Checkers and mark interfaces for debug

Maximum number of idle cycles for READY monitoring: 0 [0 - 1024]

Maximum outstanding READ Transactions per ID: 2

Maximum outstanding WRITE Transactions per ID: 2

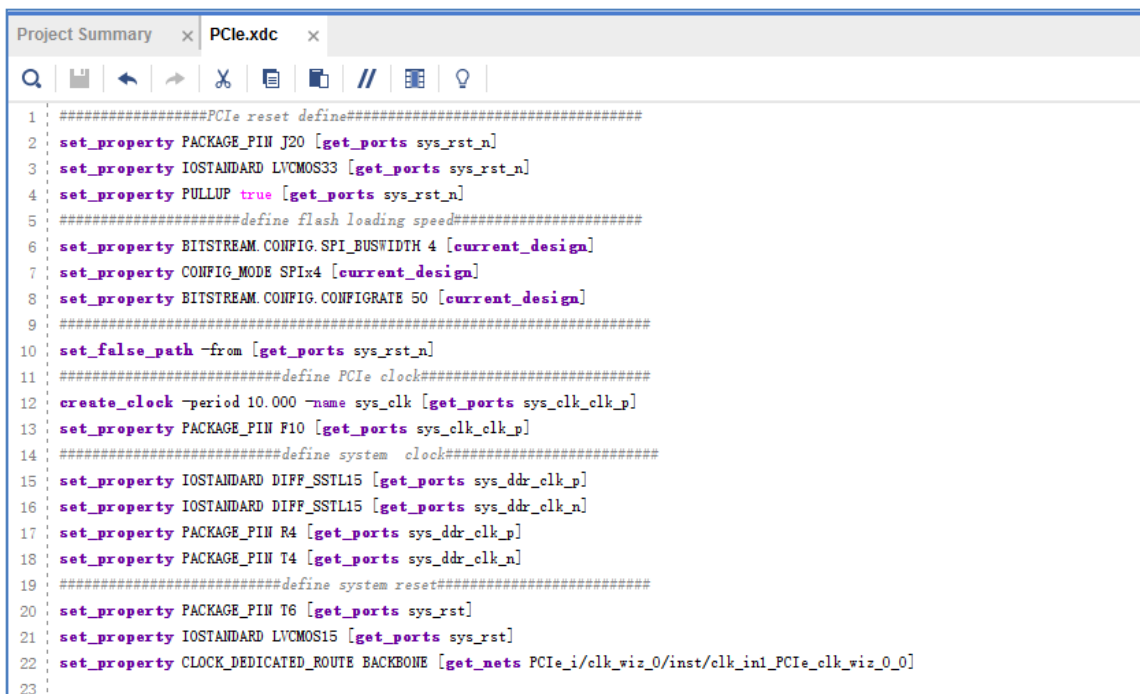
Master Interface Options

Master Interface	Secure Slave
M00: AXI	<input type="checkbox"/>

OK Cancel

至此完成了 FPGA 端的模块设置。

8. 添加顶层约束文件 XDC，具体方法在 LED 流水灯中已做介绍，在这里简单介绍工程中约束文件（PCIE.xdc），具体写法参考到 xilinx 官网下载 UG912 文档及其它约束文档，约束文件如下：

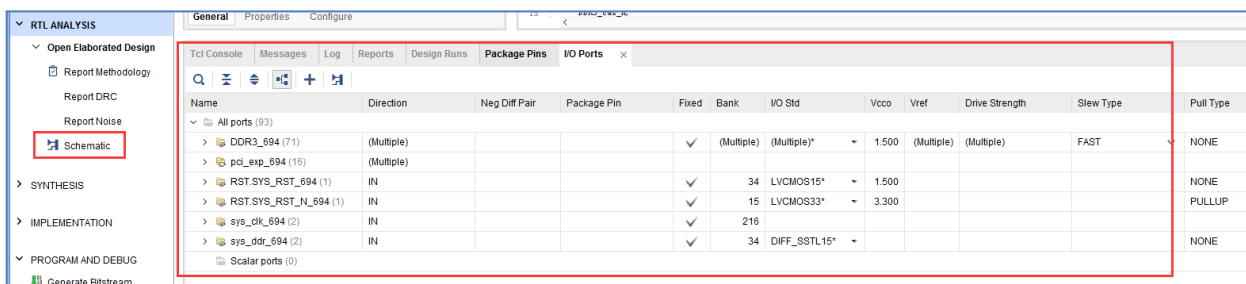


```

1 #####PCIe reset define#####
2 set_property PACKAGE_PIN J20 [get_ports sys_rst_n]
3 set_property IOSTANDARD LVCMOS33 [get_ports sys_rst_n]
4 set_property PULLUP true [get_ports sys_rst_n]
5 #####define flash loading speed#####
6 set_property BITSTREAM.CONFIG.SPI_BUSWIDTH 4 [current_design]
7 set_property CONFIG_MODE SPIx4 [current_design]
8 set_property BITSTREAM.CONFIG.CONFIGRATE 50 [current_design]
9 #####
10 set_false_path -from [get_ports sys_rst_n]
11 #####define PCIe clock#####
12 create_clock -period 10.000 -name sys_clk [get_ports sys_clk_clk_p]
13 set_property PACKAGE_PIN F10 [get_ports sys_clk_clk_p]
14 #####define system clock#####
15 set_property IOSTANDARD DIFF_SSTL15 [get_ports sys_ddr_clk_p]
16 set_property IOSTANDARD DIFF_SSTL15 [get_ports sys_ddr_clk_n]
17 set_property PACKAGE_PIN R4 [get_ports sys_ddr_clk_p]
18 set_property PACKAGE_PIN T4 [get_ports sys_ddr_clk_n]
19 #####define system reset#####
20 set_property PACKAGE_PIN T6 [get_ports sys_rst]
21 set_property IOSTANDARD LVCMOS15 [get_ports sys_rst]
22 set_property CLOCK_DEDICATED_ROUTE BACKBONE [get_nets PCIe_i/clk_wiz_0/inst/clk_in1_PCIe_clk_wiz_0_0]
23

```

当然也可以按照如下图操作进行约束，可以约束位置、电平标准，速率等属性。

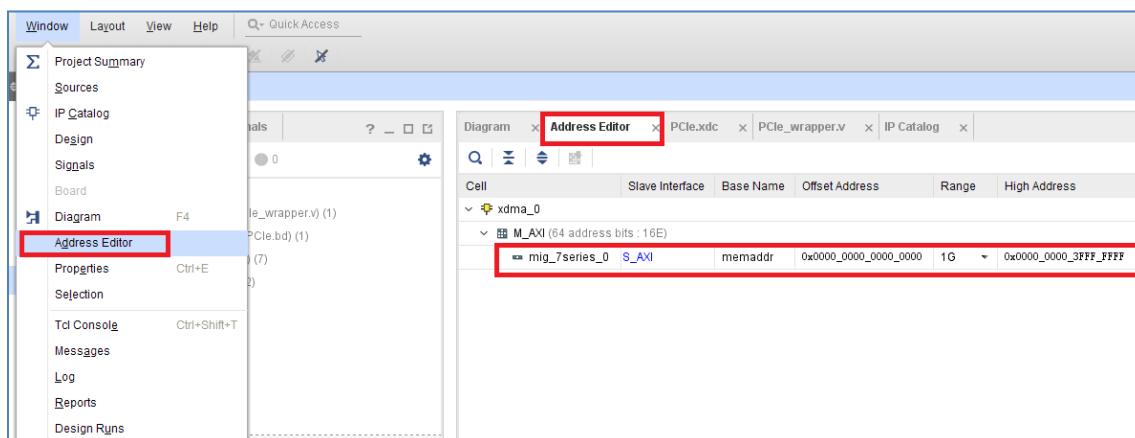


外部接口信号定义

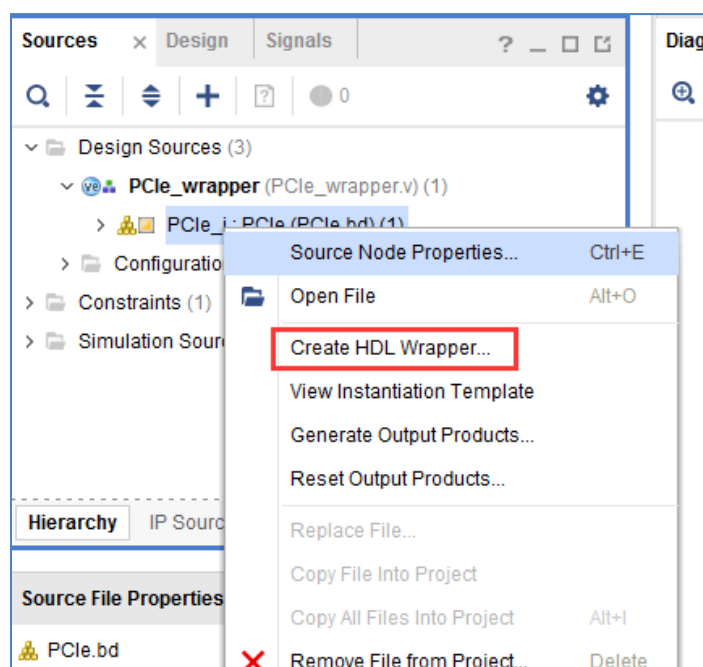
接口	描述
sys_rst_n	PCIe 复位信号
sys_ddr	系统时钟
sys_clk_clk_n、sys_clk_clk_p	PCIe 参考时钟

sys_rst	系统复位信号
---------	--------

9. 然后在如下图中对 xdma 对于 DDR3 的映射地址进行如下分配：

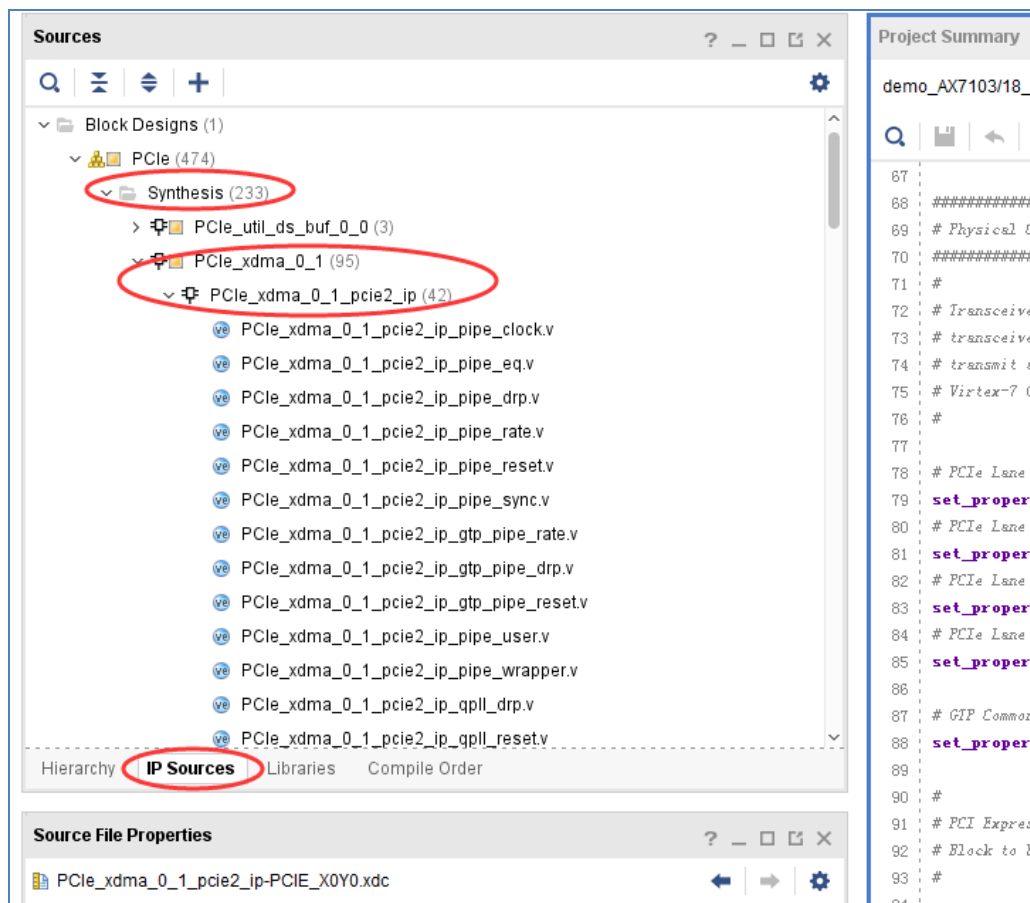


10. 完成上述步骤后，右键单击 PCIe.bd 模块选择如下图中“Create HDL Wrapper...”的进行顶层文件创建：

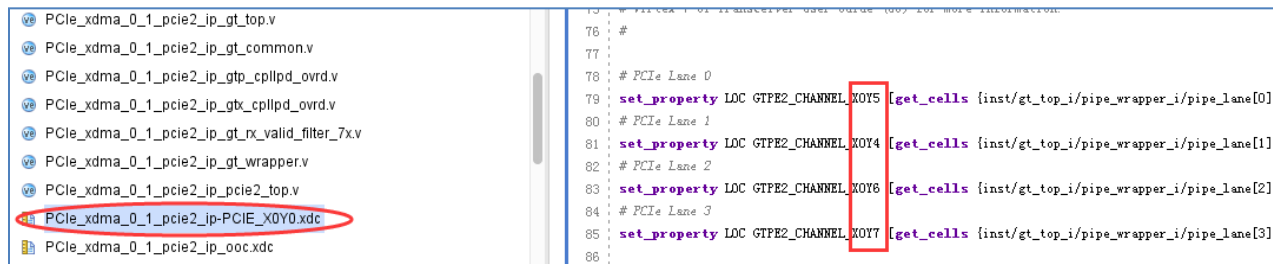


11. 编译综合下载到 AX7103 的 FLASH 之前，还需做最后一个步工作，修改 XDMA IP 的 PCIe 管脚分配约束，该约束说明适应于 AX7103 开发板的 PCIe 所有例程。参考如下：

由于 vivado 软件生成的 XDMA IP PCIe 管脚约束是默认的，与 AX7103 开发板的硬件不符，需自己修改 PCIe 管脚约束。打开 PCIe 相关例程，这里以 PCIe_test 进行说明，按照红色框提示找到 PCIe_xdma IPcore，如下图：



在打开的文件清单中找到下图中红色标注的.xdc PCIe 管脚约束文件并打开，按右侧方式修改。由于是 read only 属性，可用 uedit 软件进行修改或在顶层.xdc 文件中进行约束，本教程中用的 uedit 软件进行修改，修改方法:在文件夹中找到该文件用 uedit 软件修改即可。这种方法每改动 XDMA 一次需重新分配（修改的地方如下图所示）。

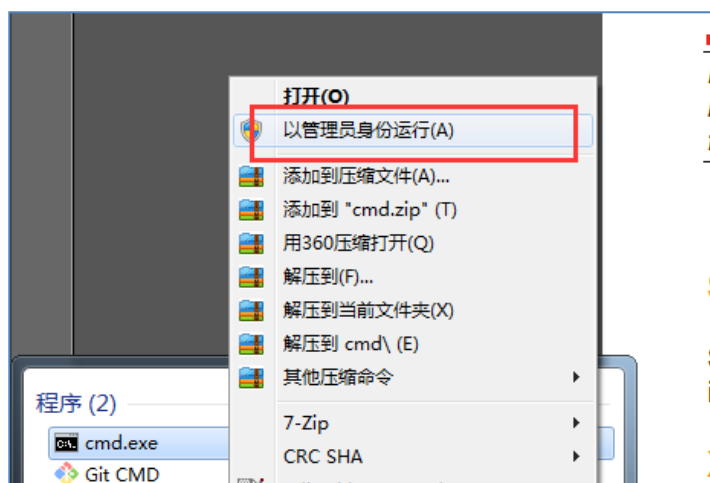


修改完成后进行编译综合，然后下载 BIN 文件到 AX7103 的 FLASH 中。把开发板插入计算机 PCIe 插槽（断电操作）。这里需要电脑主板有 x4,x8 或者 x16 的 PCIe 插槽，及 PCIe 插槽支持 PCIe 2.0 标准，具体用户可以查看主板型号来知道 PCIe 插槽的信息。

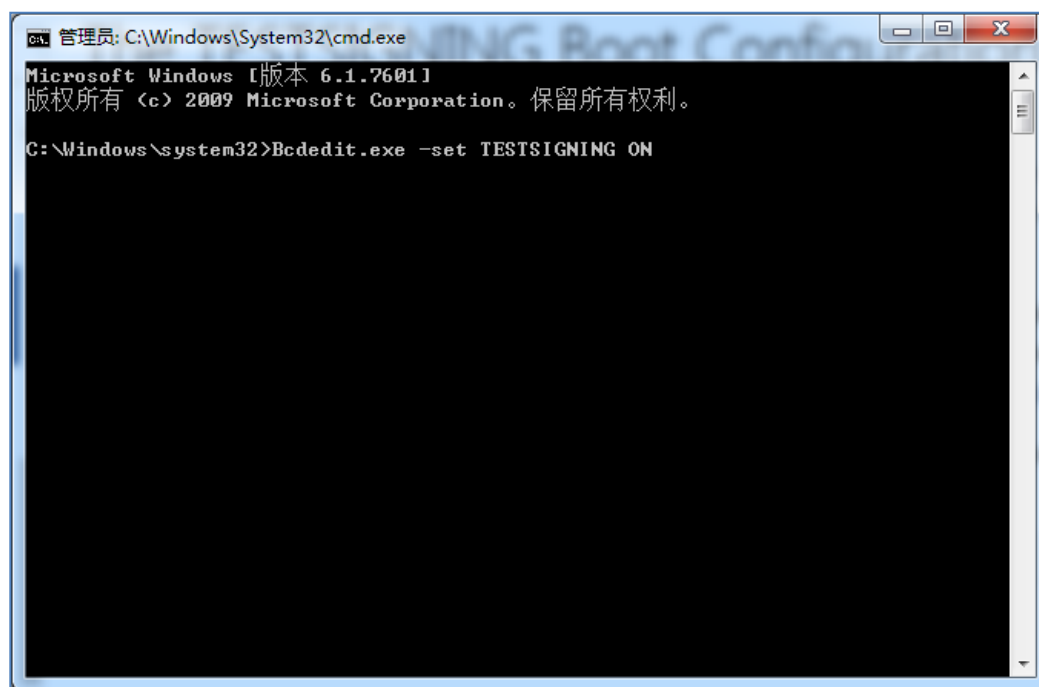
3.2 PCIe 驱动安装

由于 PCIe 驱动程序没有经过微软数字签名，所以只能用于测试，需要将系统设置到测试模式。

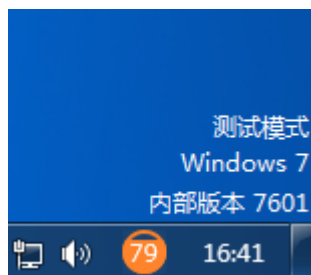
- 1) 使用管理员身份运行 cmd



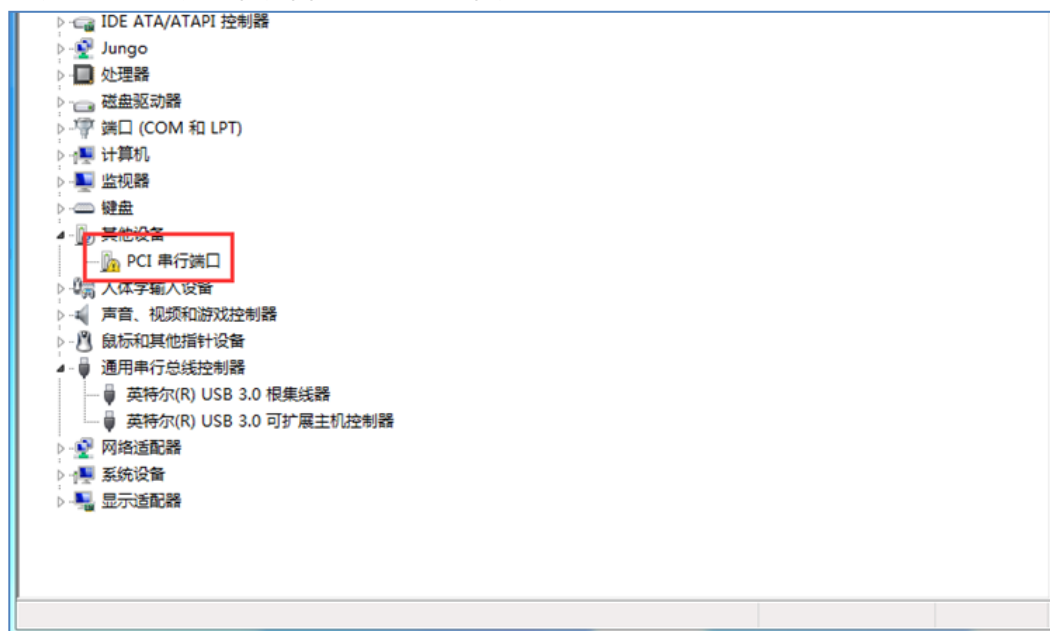
- 2) 输入命令 Bcdedit.exe -set TESTSIGNING ON 打开测试模式



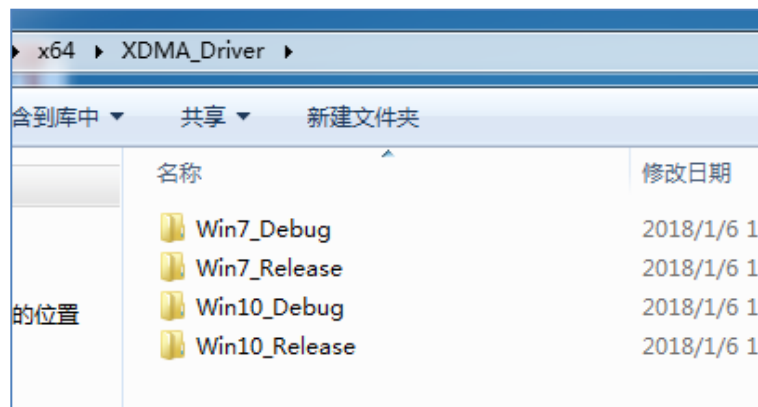
3) 重启电脑后桌面显示正在运行测试模式



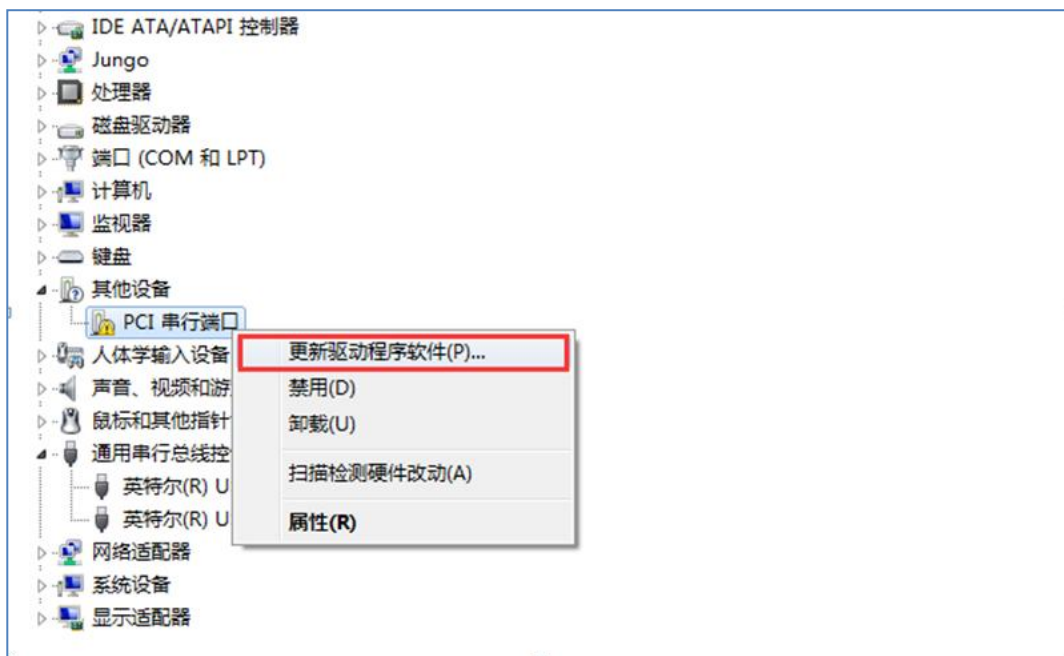
4) 没有安装 PCIE 驱动时，设备管理器如下，发现一个 PCI 串行端口。



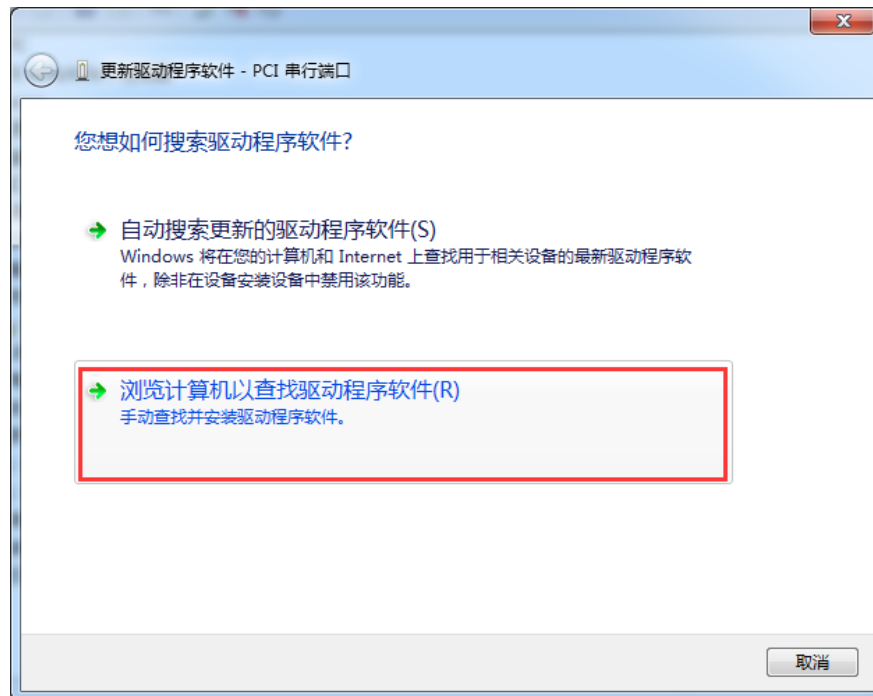
5) 例程提供的编译好的驱动程序，在工程下的 driver 目录下（根据操作系统选择相应驱动），在 demo 目录下的“xilinx PCIe 驱动相关资料”提供了 xilinx 官方的 Windows 和 linux 驱动源码及测试程序，有兴趣的自己可以研究研究：



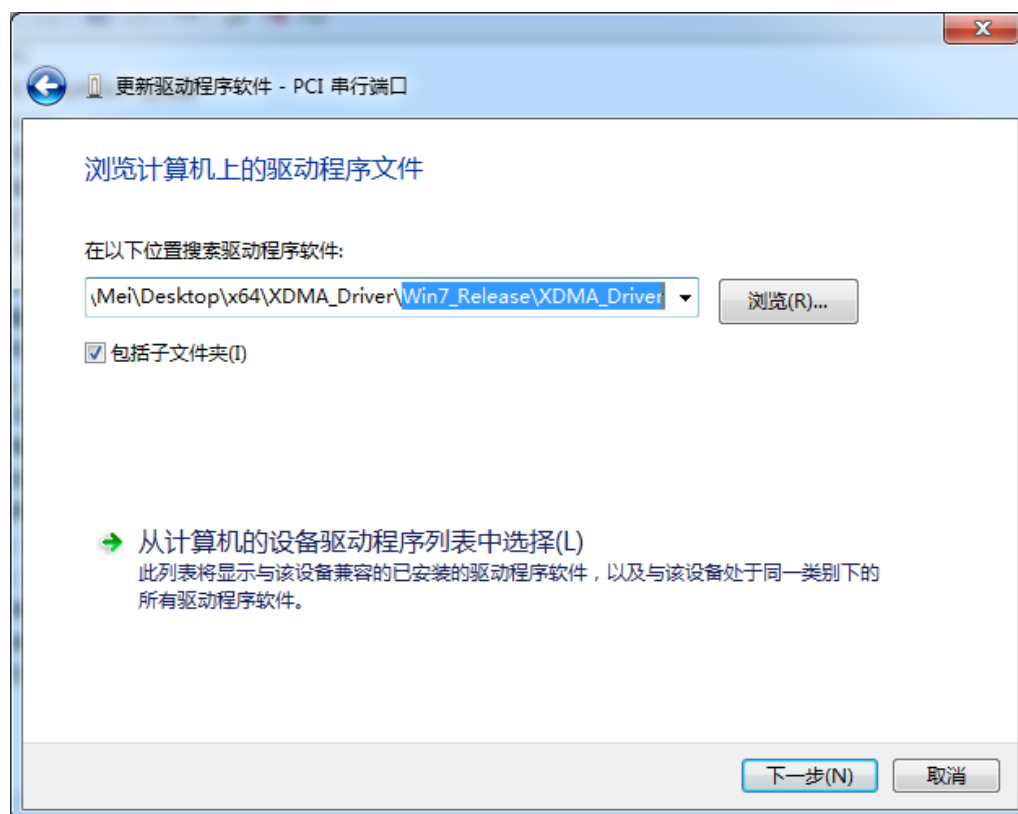
6) 选择设备，右键更新驱动



7) 浏览计算机以查找驱动程序软件



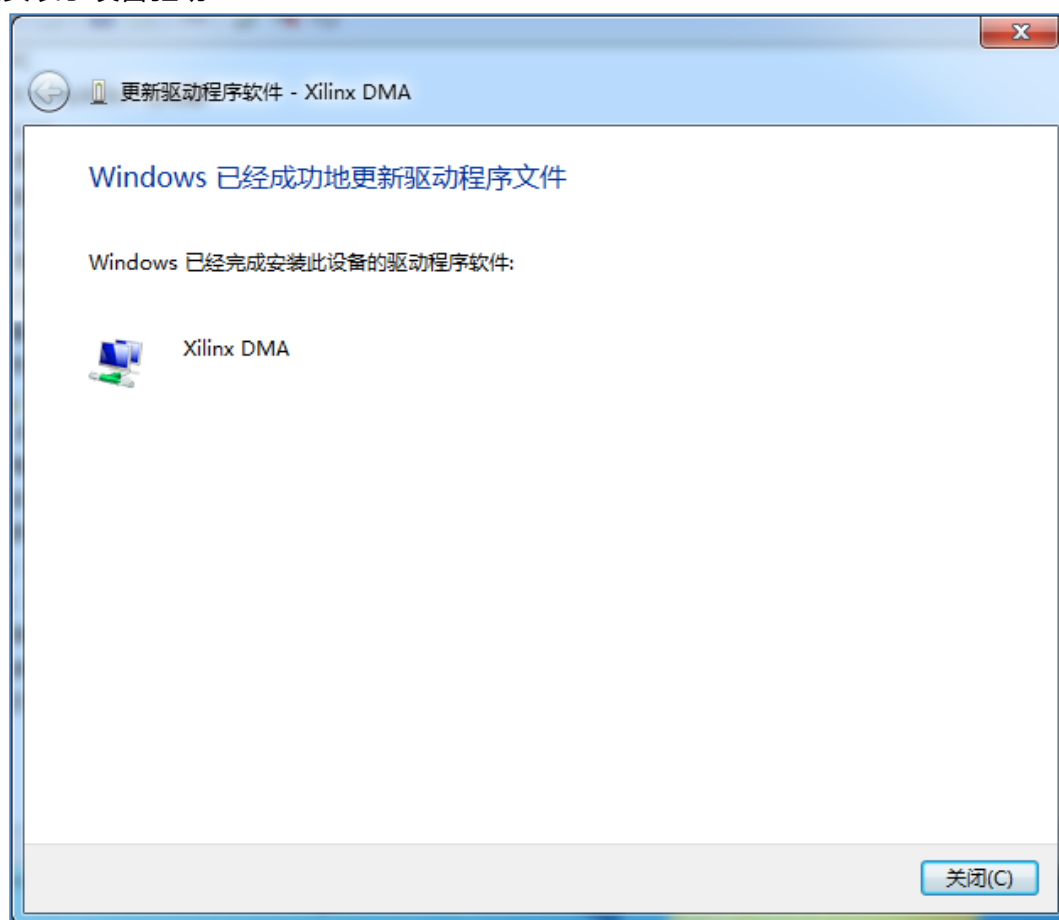
8) 测试电脑安装的是 win7，这里使用 Win7_Release 版本



9) 有个安全警告，选择安装



10) 正确安装了设备驱动

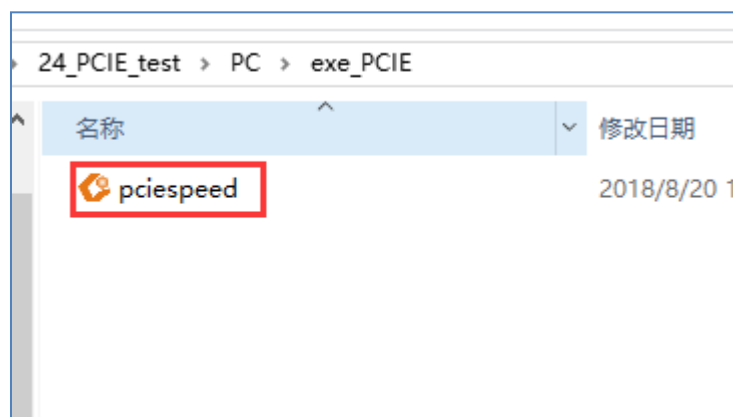


3.3 上位机测试程序

说明：上位机测试程序的开发平台为 QT5.6.2，提供了测试源代码。文中不进行简介。

4 实验现象

开启上位机测程序进行 PCIe 速度测试，打开下图的测速软件 pciespeed，位于 26_PCIE_test\PC\exe_PCIE，实验结果如下：



1) 读写同时进行测试



2) 只读测试



3) 只写测试

