MYD-Y7Z020/10/007S Tutorial

V1.0

Revision History

Revision Date	Revised Contents	Version	Revised By	Reviewed By
2020.9.3	First Edition	V1.0		

Table of Contents

1 Overview

This tutorial will walk you through generating a hardware platform in Vivado 2017.4 for the Myir MYD-Y7Z020/10/007S Starter Kit and running several software applications on the board to test various interfaces.

2 Objectives

At the end of this tutorial you will be able to:

- At the end of this tutorial you will be able to:
- Build a hardware platform for the MYD-Y7Z020/10/007S Starter Kit in Vivado 2017.4
- Export the hardware platform to the SDK 2017.4
- Create a simple “Hello World” software application in SDK and run it on the MYD-Y7Z020/10/007S Starter Kit
- Create various application software in SDK to test several interfaces on the MYD-Y7Z020/10/007S Starter Kit
- Boot the MYD-Y7Z020/10/007S from the microSD card

3 Tutorial Requirements

This tutorial will require the following software and hardware setups.

3.1 Software

The software requirements for this reference design are:

- Xilinx Vivado 2017.4 with SDK
- USB Serial Port driver
- Putty terminal program

3.2 Hardware

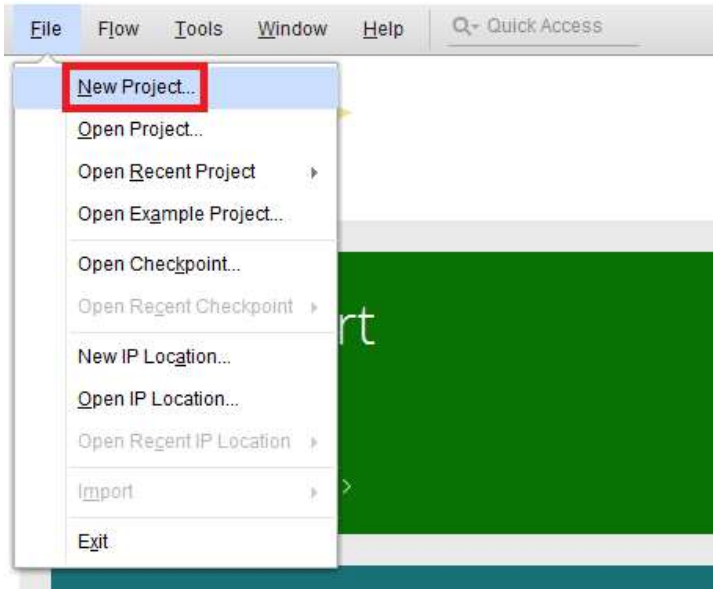
The hardware setup for this reference design is:

- Computer with 4 GB RAM and 1 GB virtual memory (recommended)
- MYD-Y7Z020/10/007S Starter Kit
- 12V power supply
- Power cable
- Ethernet cable
- MY-UART012U USB to COM cable
- Xilinx USB Platform Cable DLC9

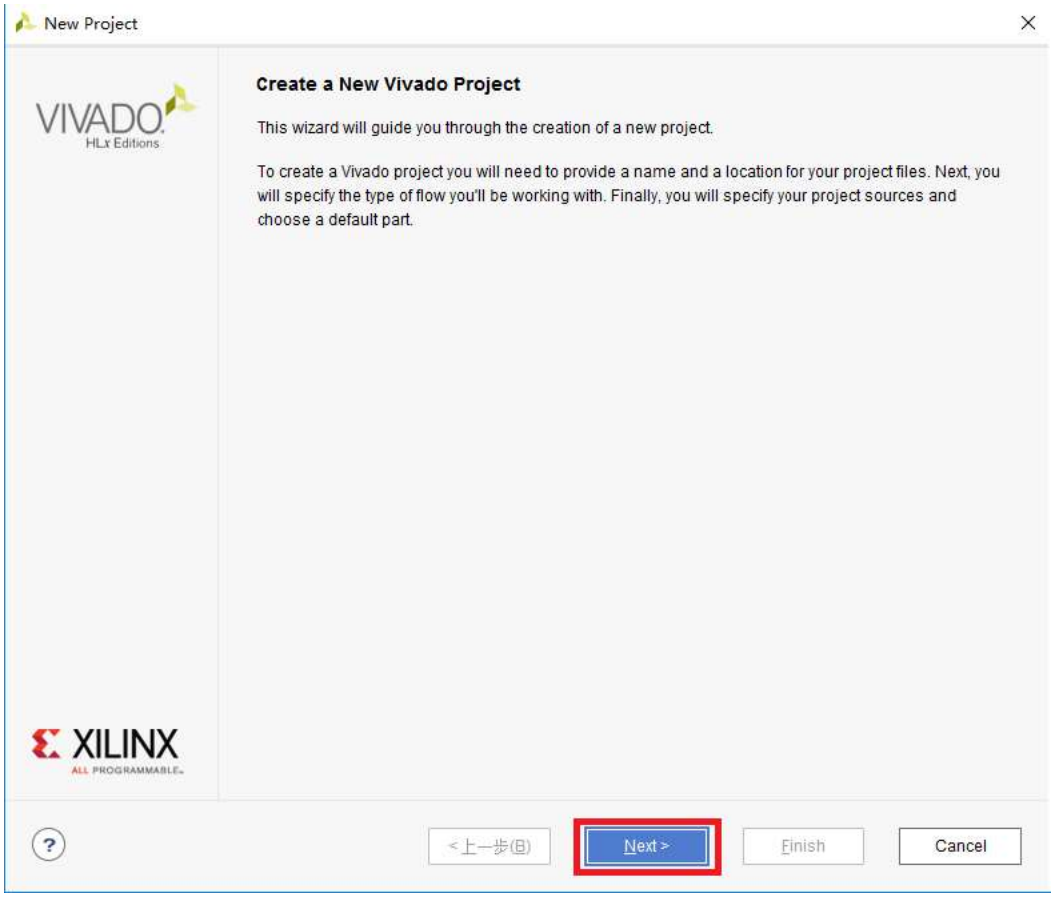
4 Creating a Hardware Platform for the MYD-Y7Z020/10/007S Starter Kit

Please follow the steps shown below to generate the hardware platform for thistutorial using Vivado 2017.4 tool.

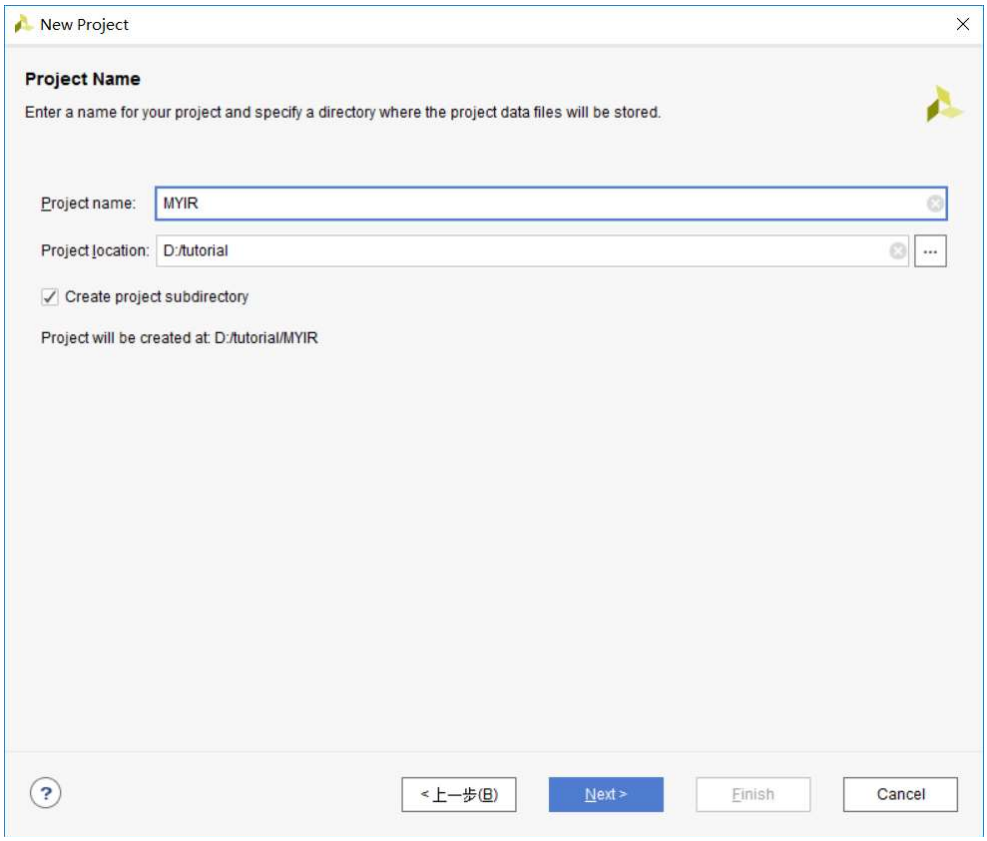
- Start the Vivado tool via Start > All Programs > Xilinx Design Tools >Vivado 2017.4>Vivado 2017.4
- Select Create New Project.



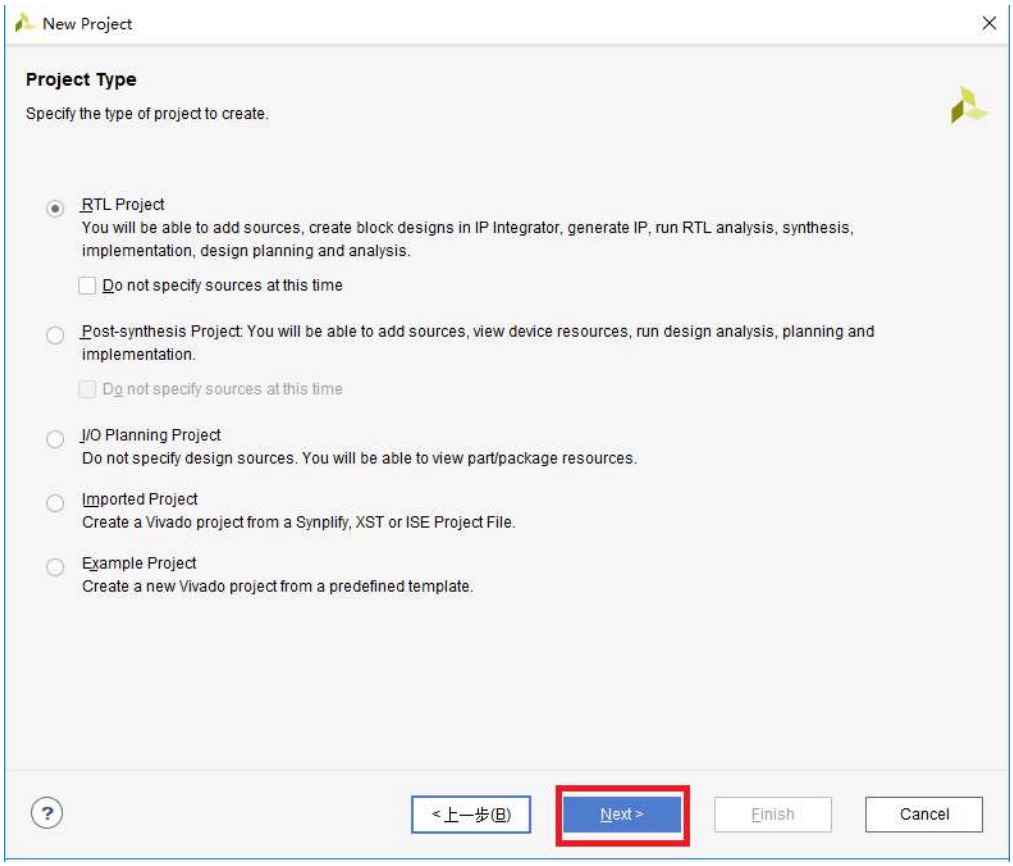
• Click **Next** to continue.



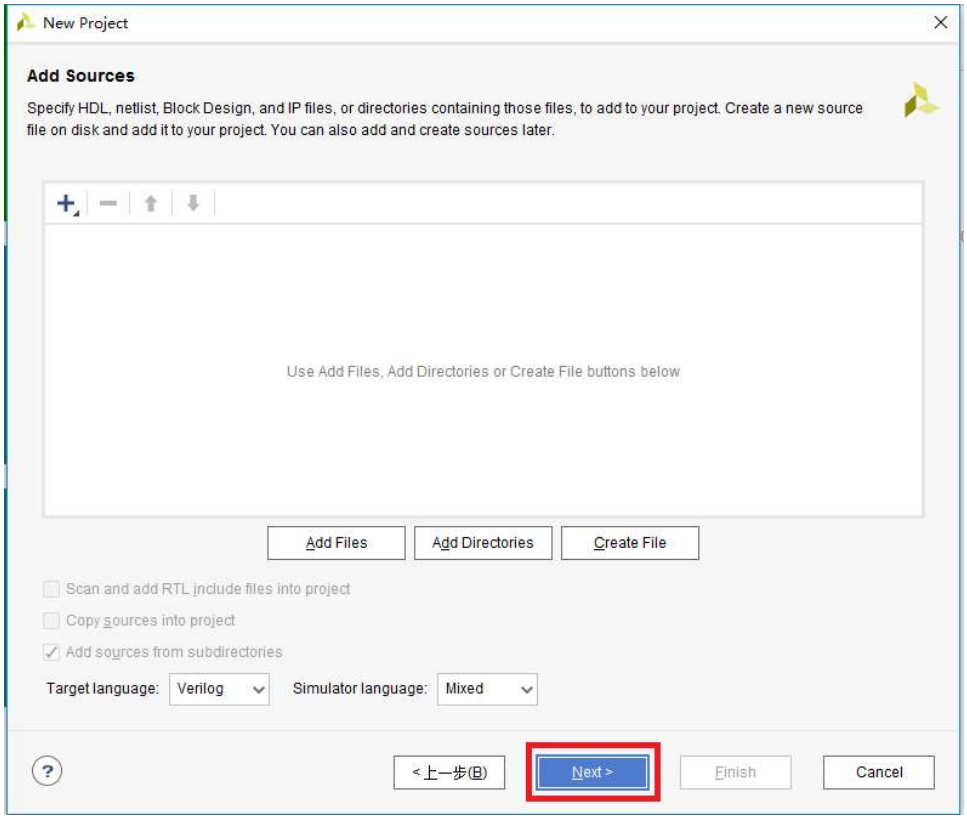
• Set the project name and location and click **Next** to continue. In this case, the project locationis set to **D:/tutorial** and the project name is set to **MYIR**. Make sure the **Create project subdirectory** box is checked as shown in the following figure.



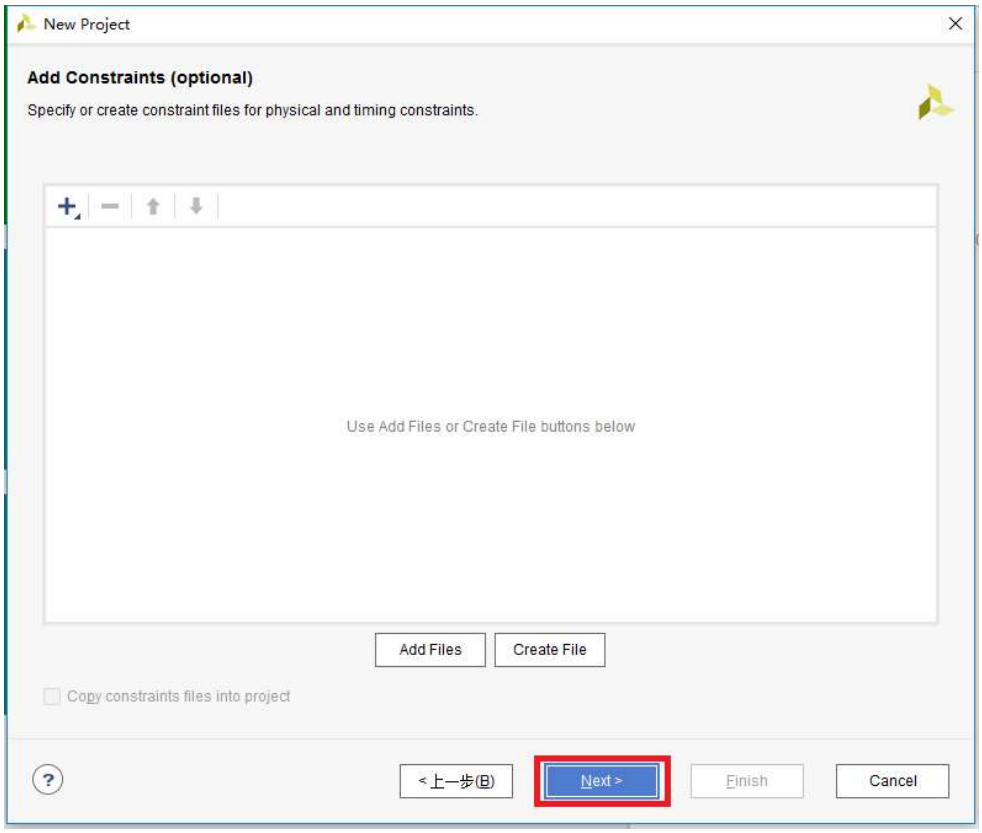
• Select the **RTL Project** type and click **Next** to continue.


New Project
Project Type
Specify the type of project to create.
RTL Project
You will be able to add sources, create block designs in IP Integrator, generate IP, run RTL analysis, synthesis, implementation, design planning and analysis.
Do not specify sources at this time
Post-synthesis Project: You will be able to add sources, view device resources, run design analysis, planning and implementation.
Do not specify sources at this time
I/O Planning Project
Do not specify design sources. You will be able to view part/package resources.
Imported Project
Create a Vivado project from a Synplify, XST or ISE Project File.
Example Project
Create a new Vivado project from a predefined template.
< 上一步(B)
Next >
Finish
Cancel

• Click **Next** to continue.



• Click **Next** to continue.



• In the **Default Part** dialog box

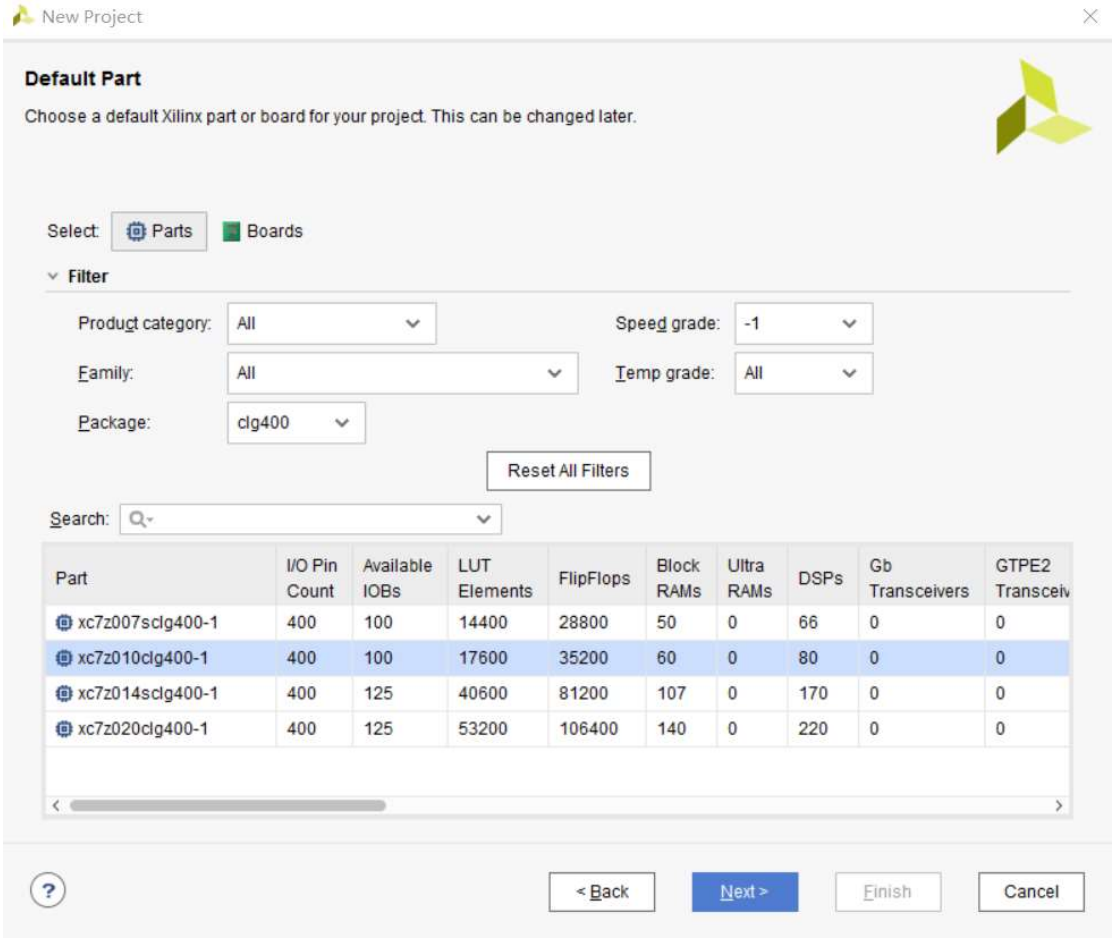
a. Click on **Parts,** next to **Select:** as shown below.

b. Set the **Speed Grade** to **-1**.

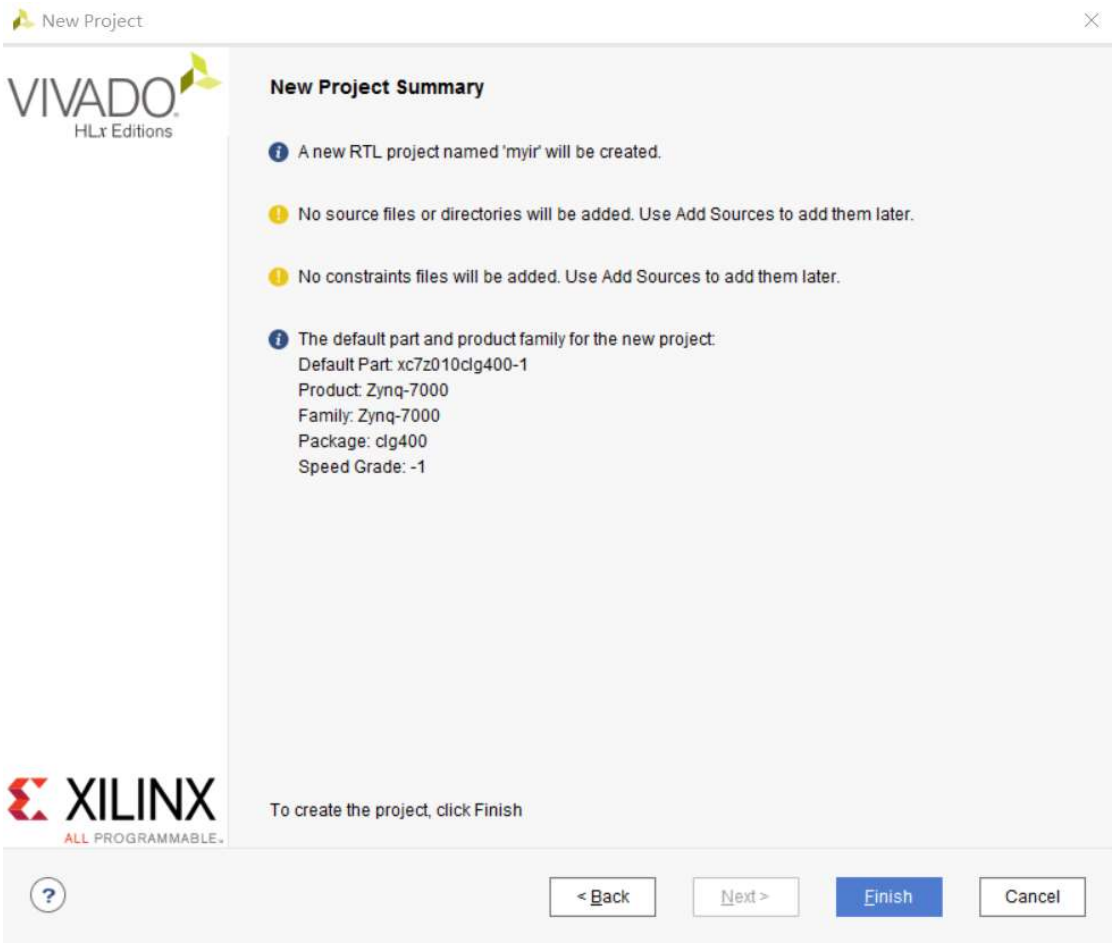
c. Set the **Package** to **clg400** Set

d. Select art **xc7z010clg400-1**.

e. Click **Next** to continue.



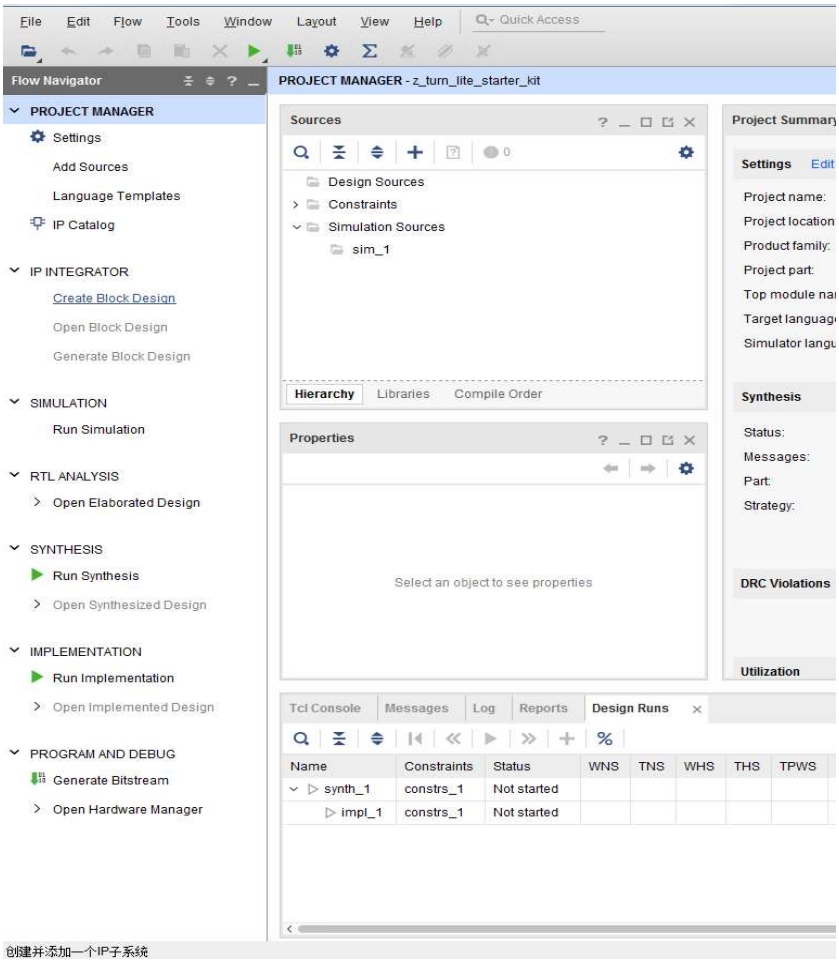
• In the **New Project Summary** dialog box, click **Finish** to continue.


New Project
VIVADO
HLx Editions
New Project Summary
A new RTL project named 'myir' will be created.
No source files or directories will be added. Use Add Sources to add them later.
No constraints files will be added. Use Add Sources to add them later.
The default part and product family for the new project:
Default Part: xc7z010clg400-1
Product: Zynq-7000
Family: Zynq-7000
Package: clg400
Speed Grade: -1
XILINX
ALL PROGRAMMABLE.
To create the project, click Finish
< Back
Next >
Finish
Cancel

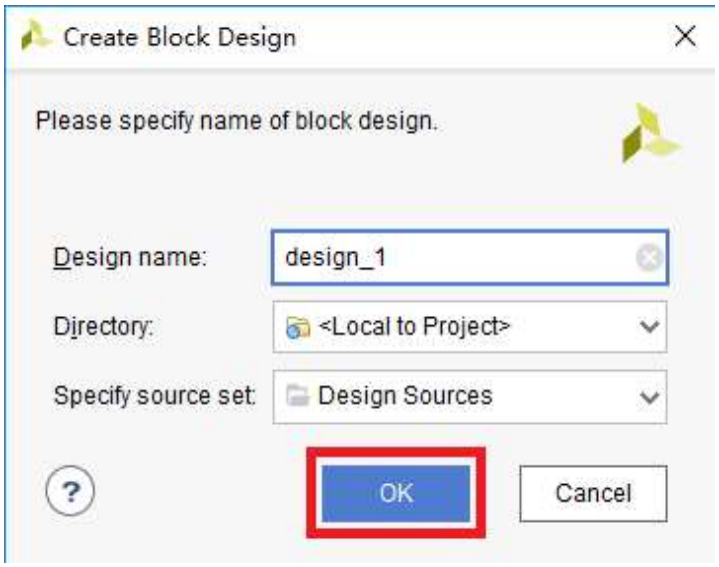
4.1 Creating a Block Design

In this section we will be creating a Block Design for the hardware platform. Once the Block Design canvas is created, IP cores can be added to the design form the Vivado **IP Catalog**.

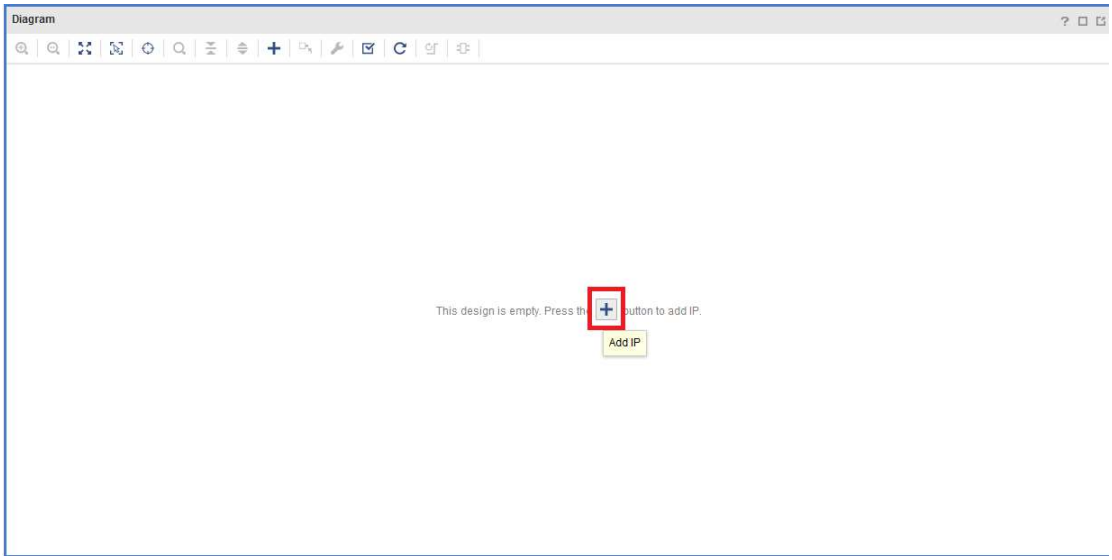
• Click on the **Create Block Design** as shown in the following figure.



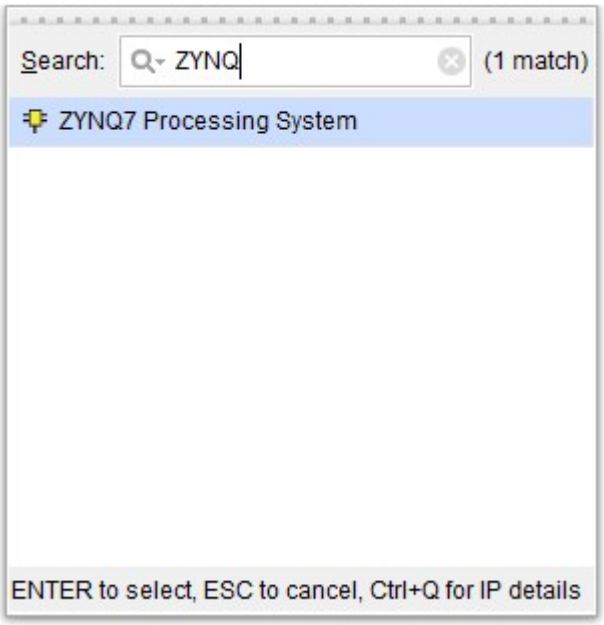
• When the following dialog box appears, click **OK** to continue. You may change the default **design_1** name to anything you wish. We will be using the default name for this tutorial.


Create Block Design
Please specify name of block design.
Design name:
design_1
Directory:
<Local to Project>
Specify source set:
Design Sources
OK
Cancel

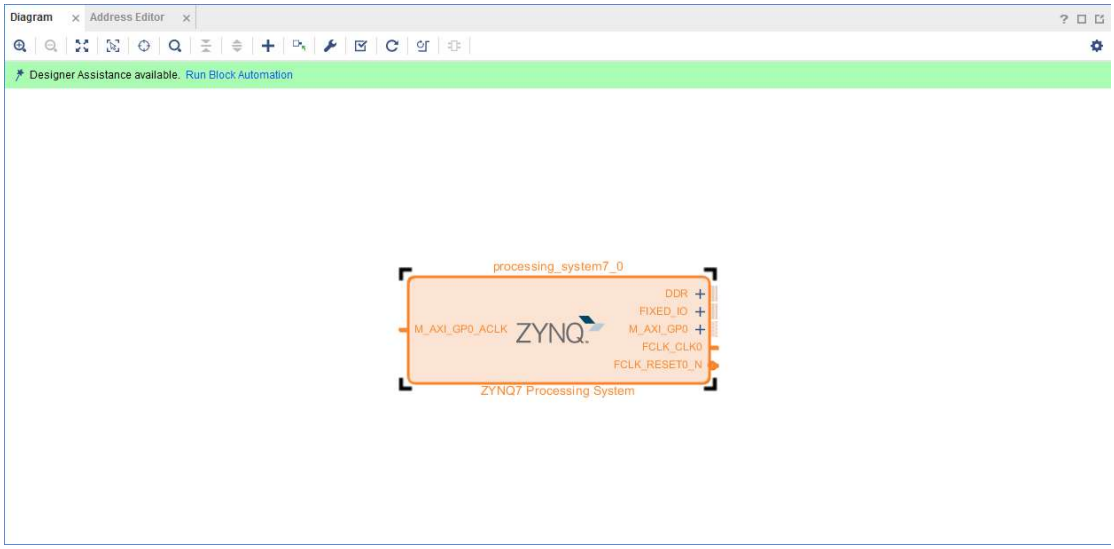
• Click on the **Add IP** icon in the white canvas area as shown in the following figure to begin adding IP cores to the design.



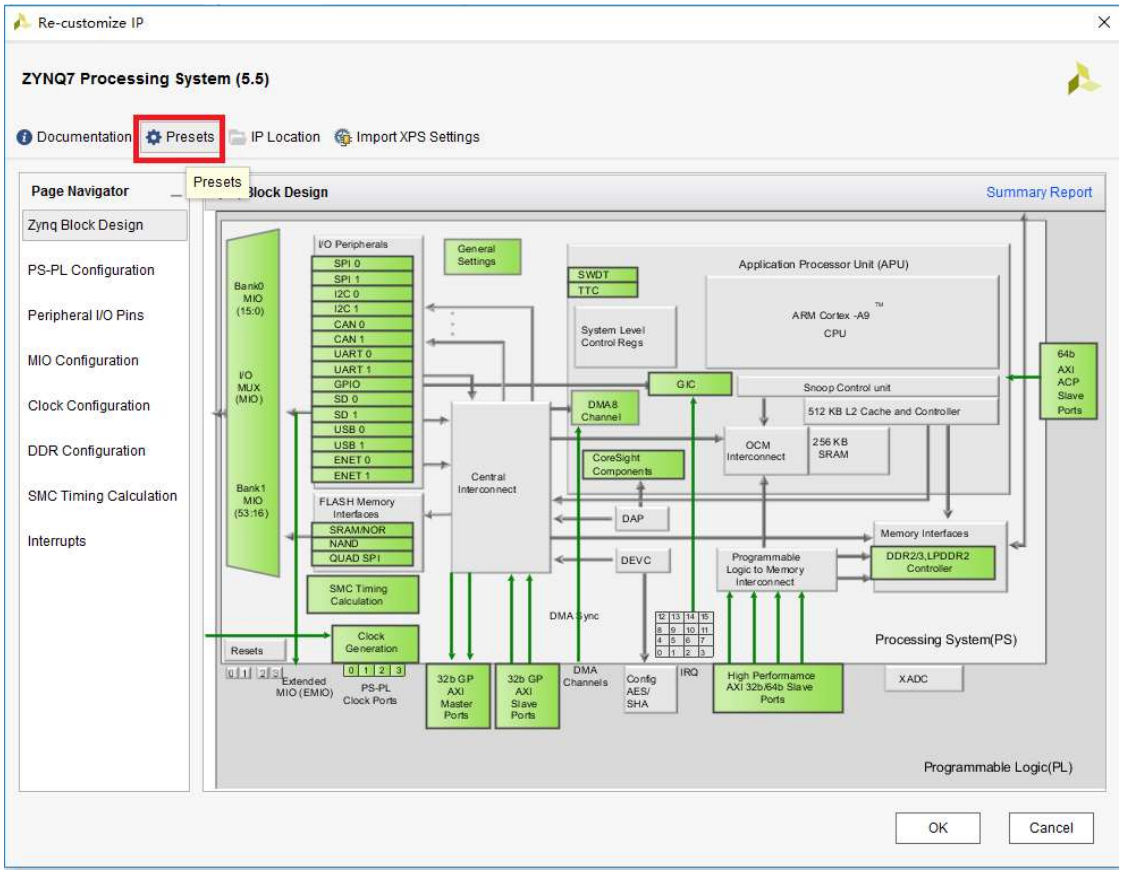
• The first step is to add the Zynq IP core to the design. Type **Zynq** in the **Search** box and then double-click on the **ZYNQ7 Processing System** as shown in the following figure.



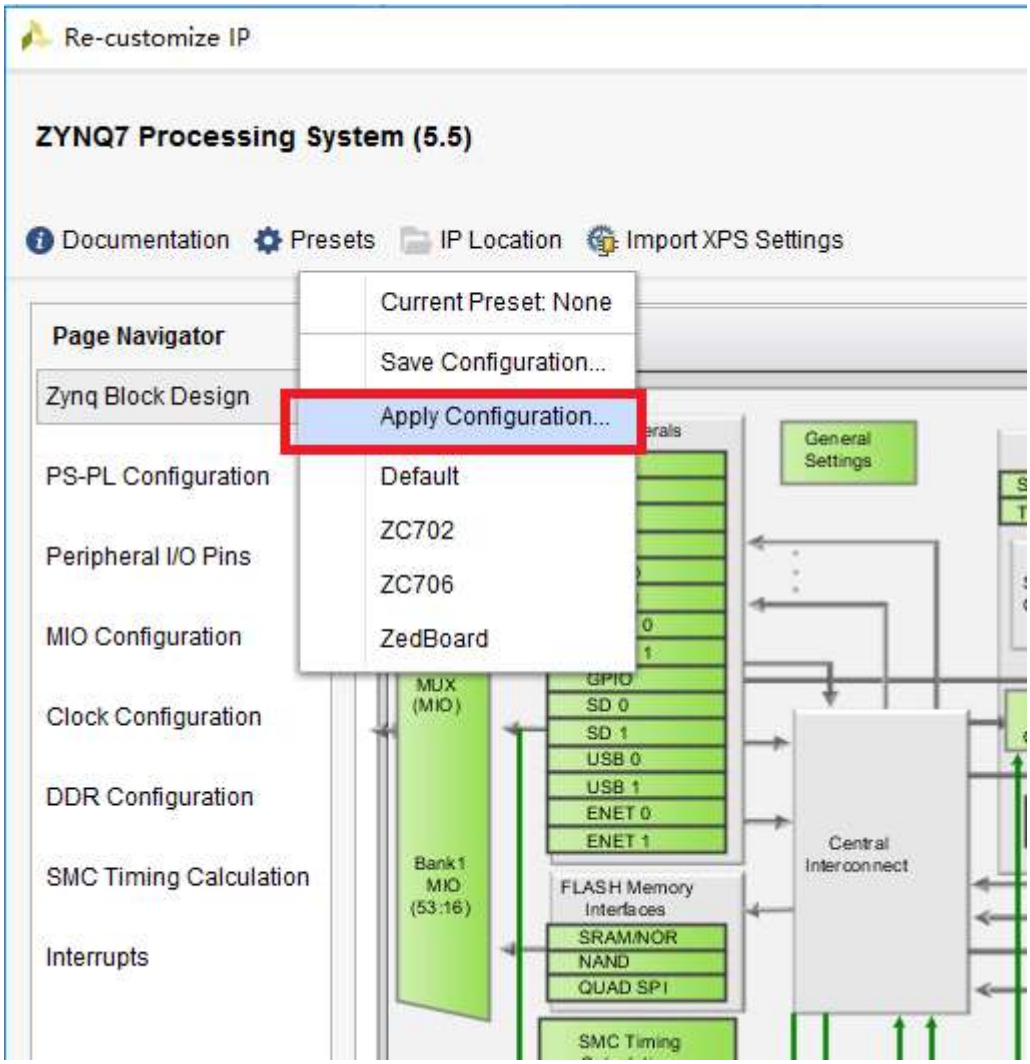
The Zynq IP will be added to the design as shown in the following figure.



• Double-click **ZYNQ IP** and click **Presets**.

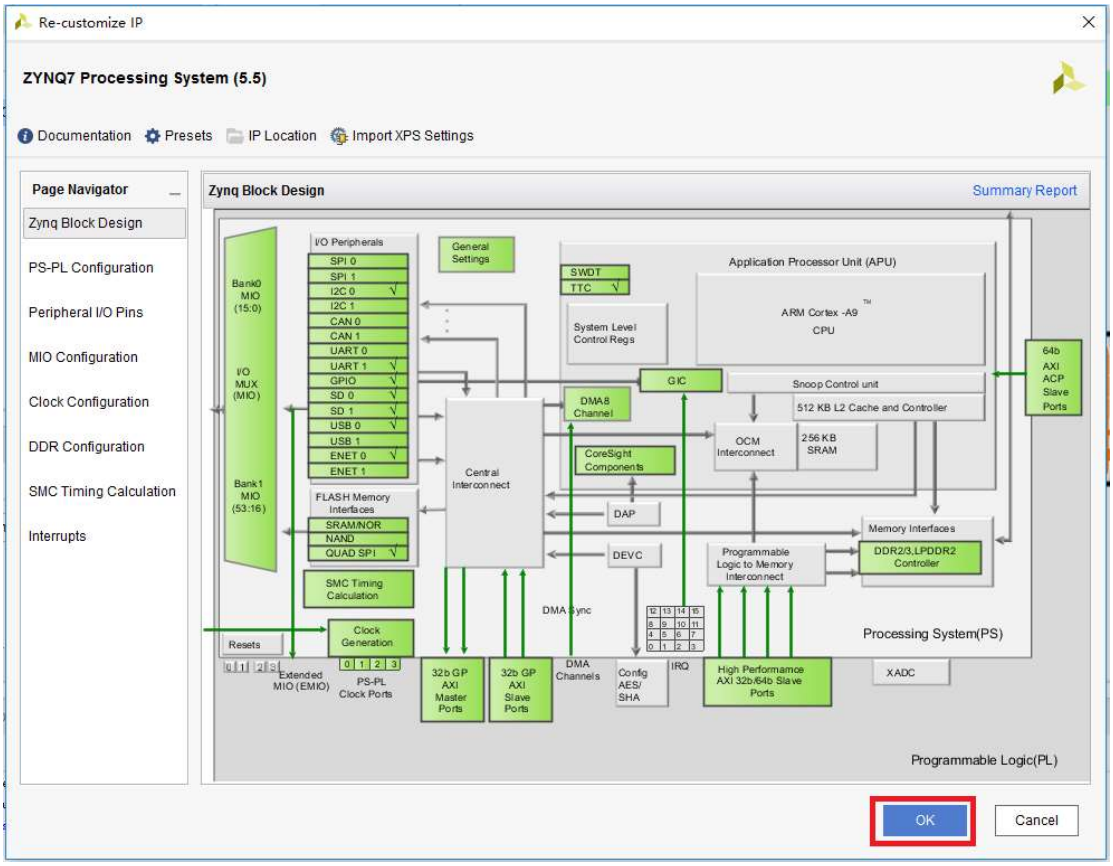


• Select Applying configuration

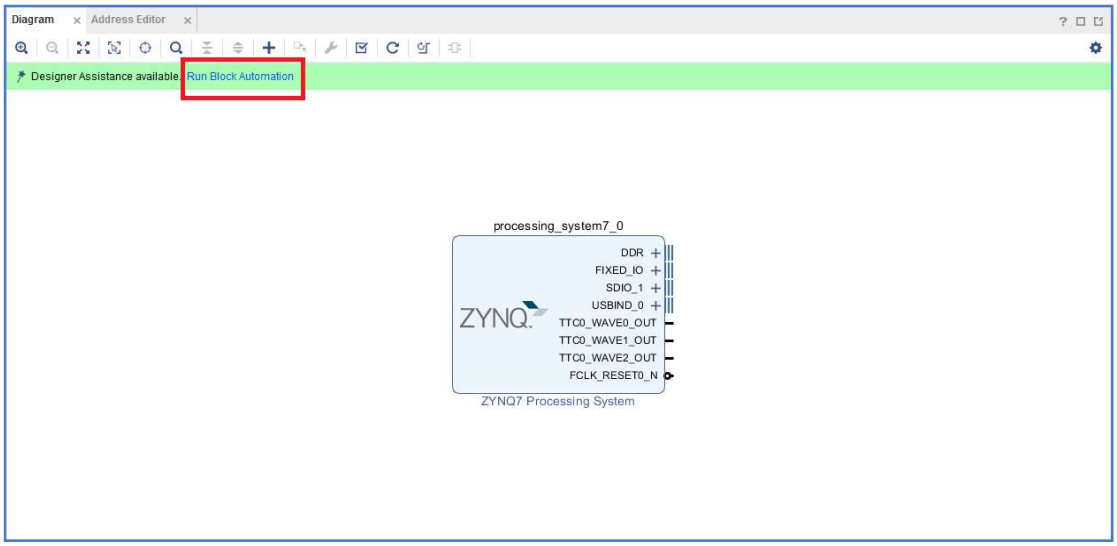


• Select **MYIR.tcl** and click **OK**.
(PATH: *\05-Programmable_Logic\7z010\)

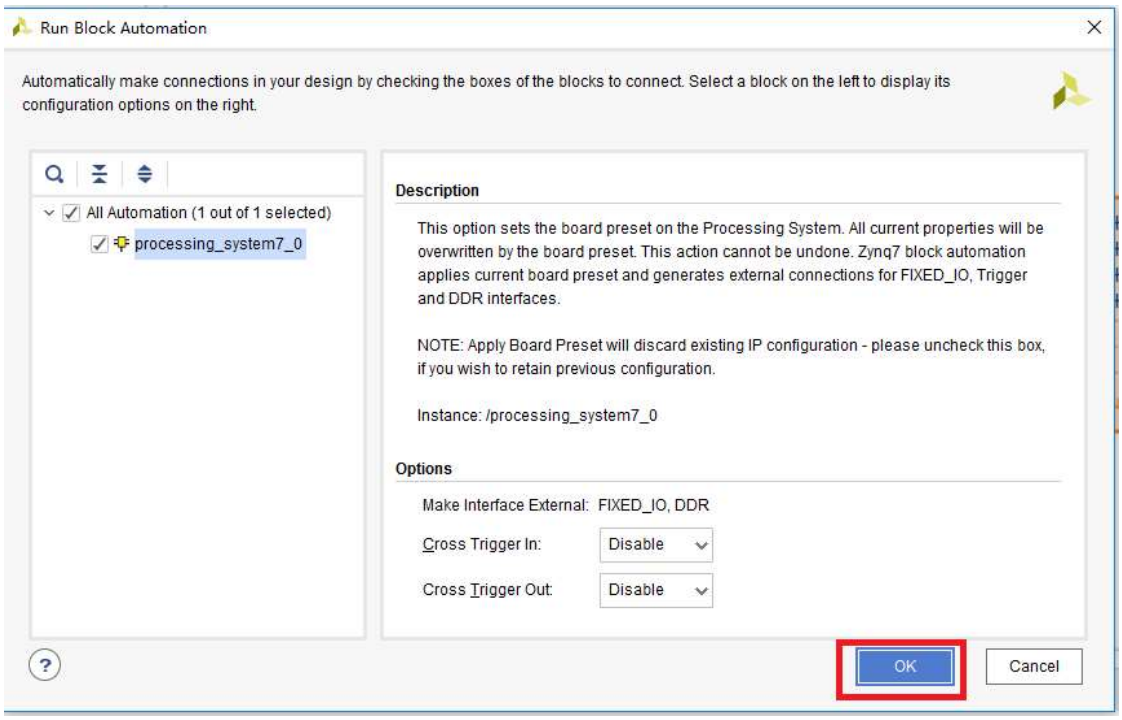
• Click **OK**.



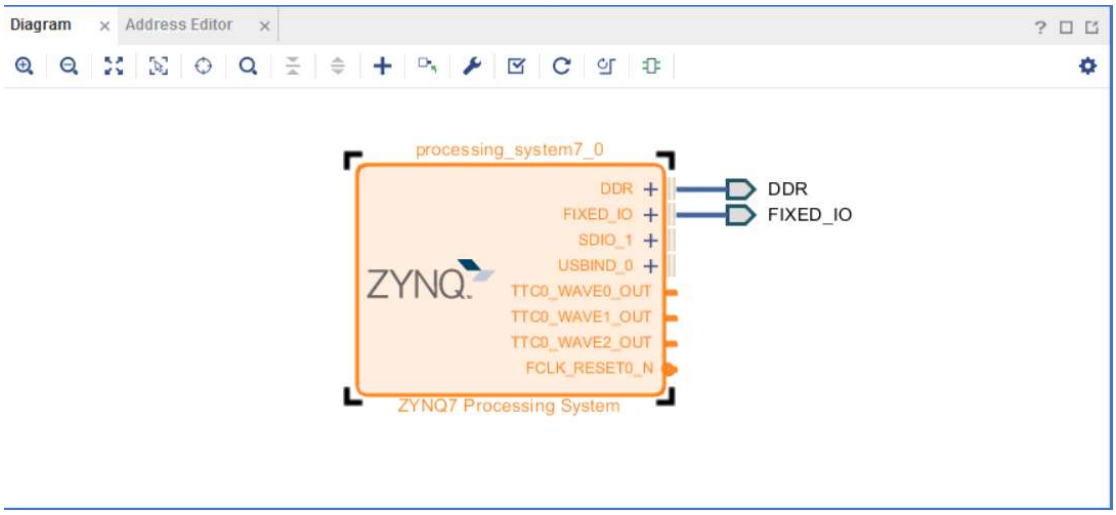
• The Zynq IP will be modified as shown in the following figure.The Zynq device has not yet been configured. Click on the **Run Block Automation** to configure the Zynq device with the IO settings.



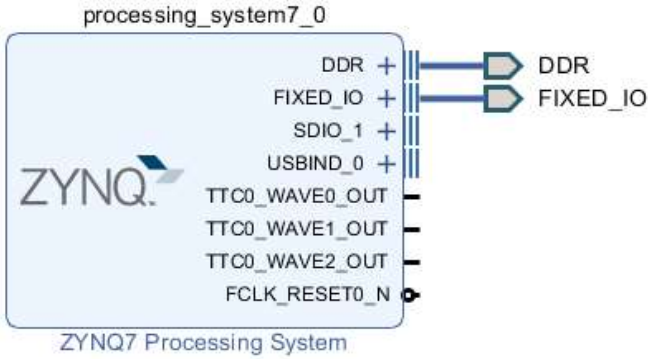
• When the following dialog box appears, click **OK** to continue.



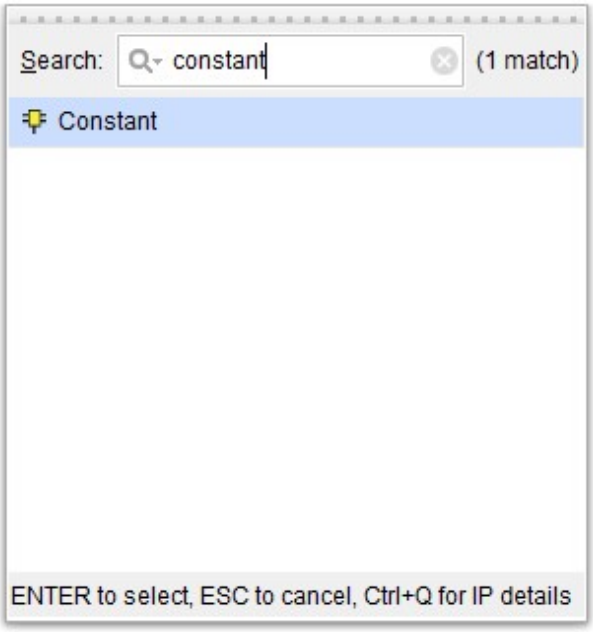
• The Zynq device will be configured as shown in the following figure. The Zynq device is now configured with the board level settings such as PS DDR3, PS peripheral selections, clocking, etc.



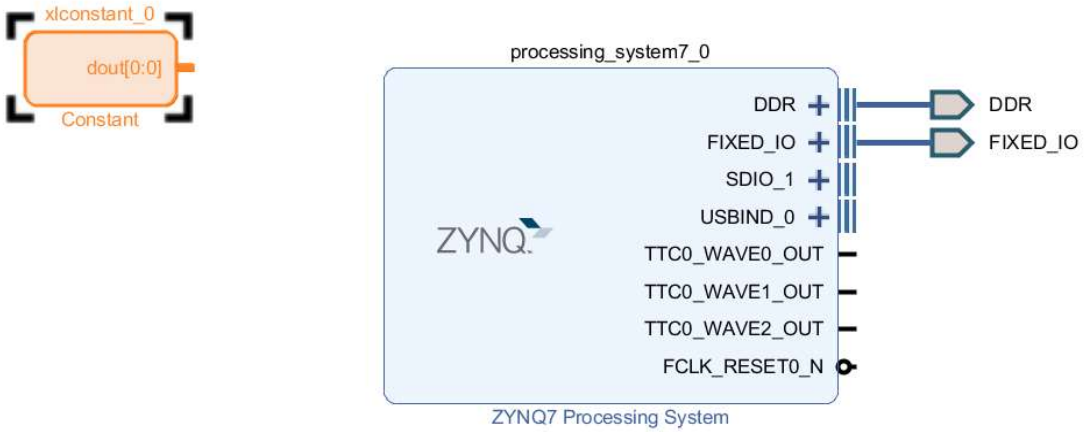
Click **Add IP**



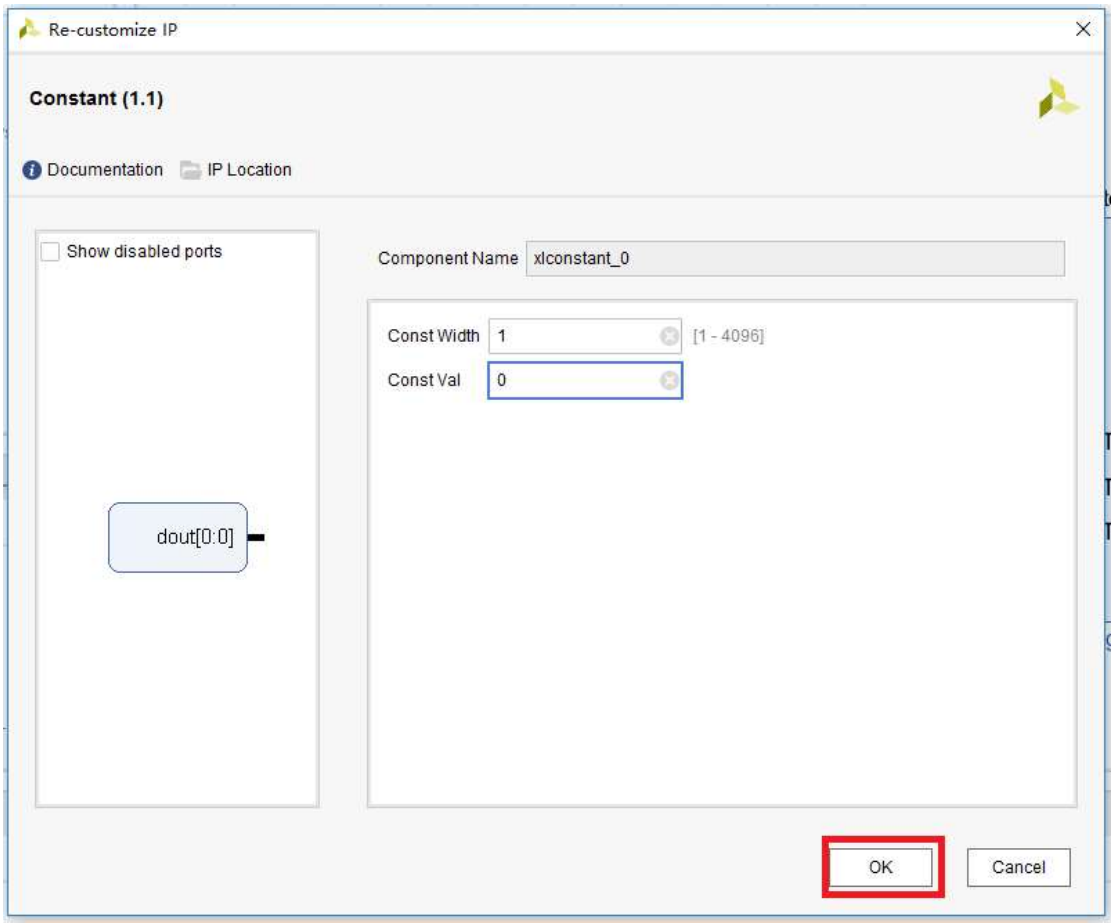
Type in **constant**, double click**Constant**



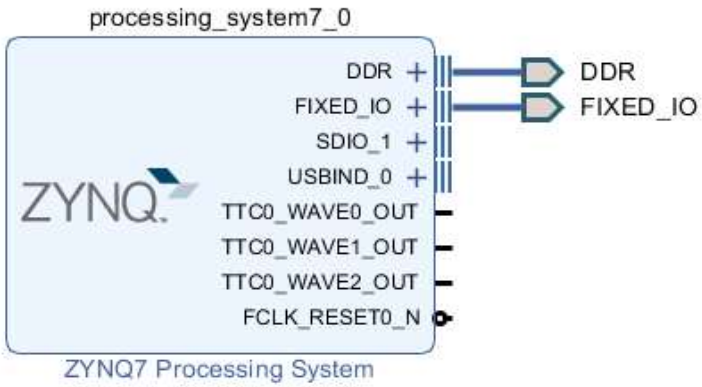
The Constant IP will be added to the design as shown in the following figure.

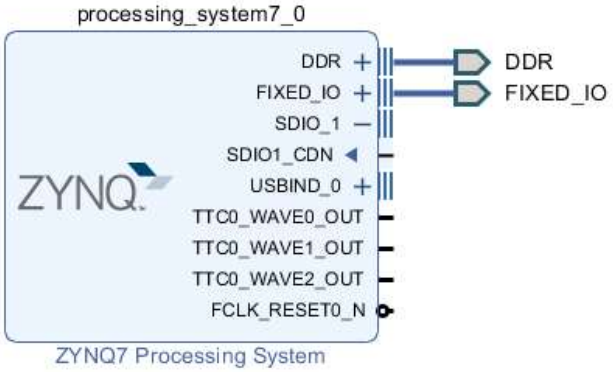


Double-click **Constant IP** to configure, modify **Const Val** to **0** and click **OK**.

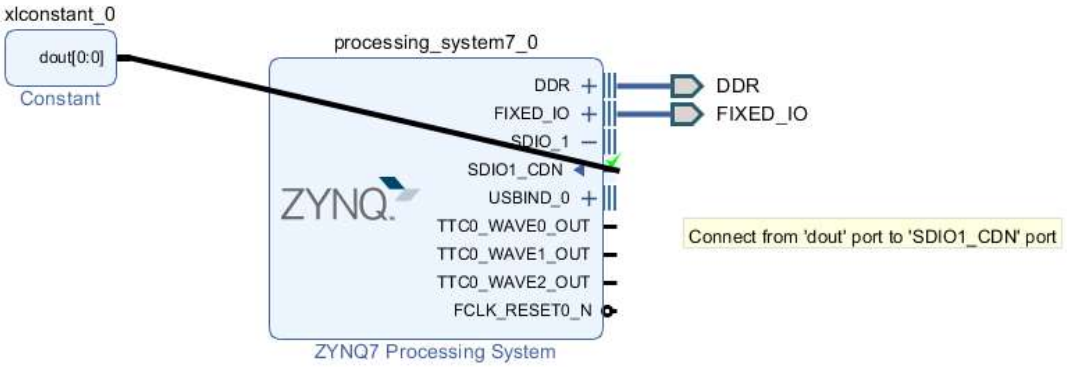


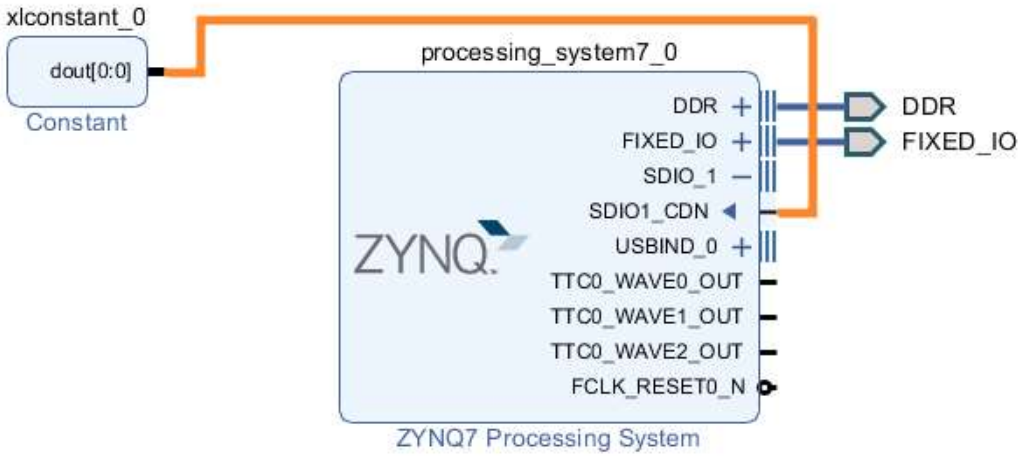
Click + right side **SDIO_1 .**



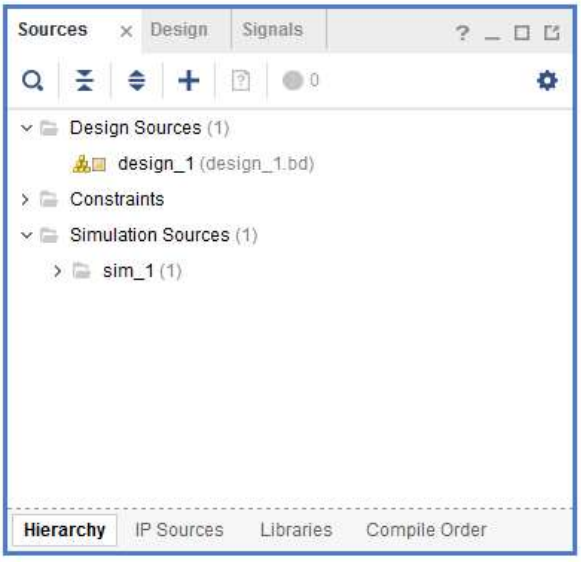


Wiringdout[0:0] to SDIO1_CDN.

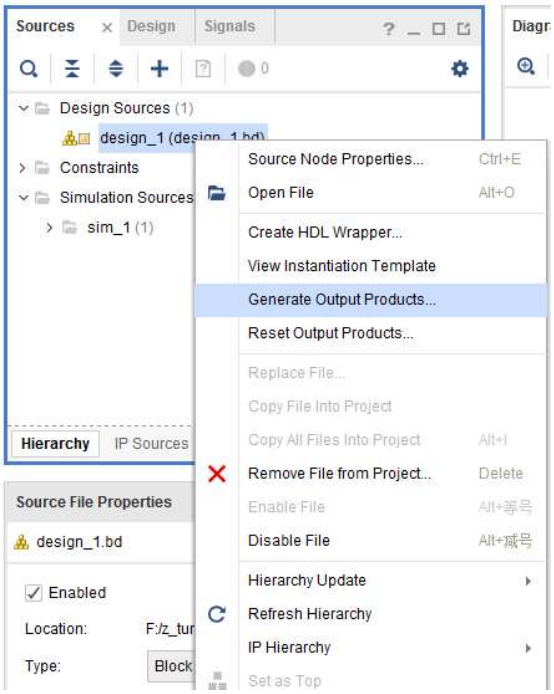




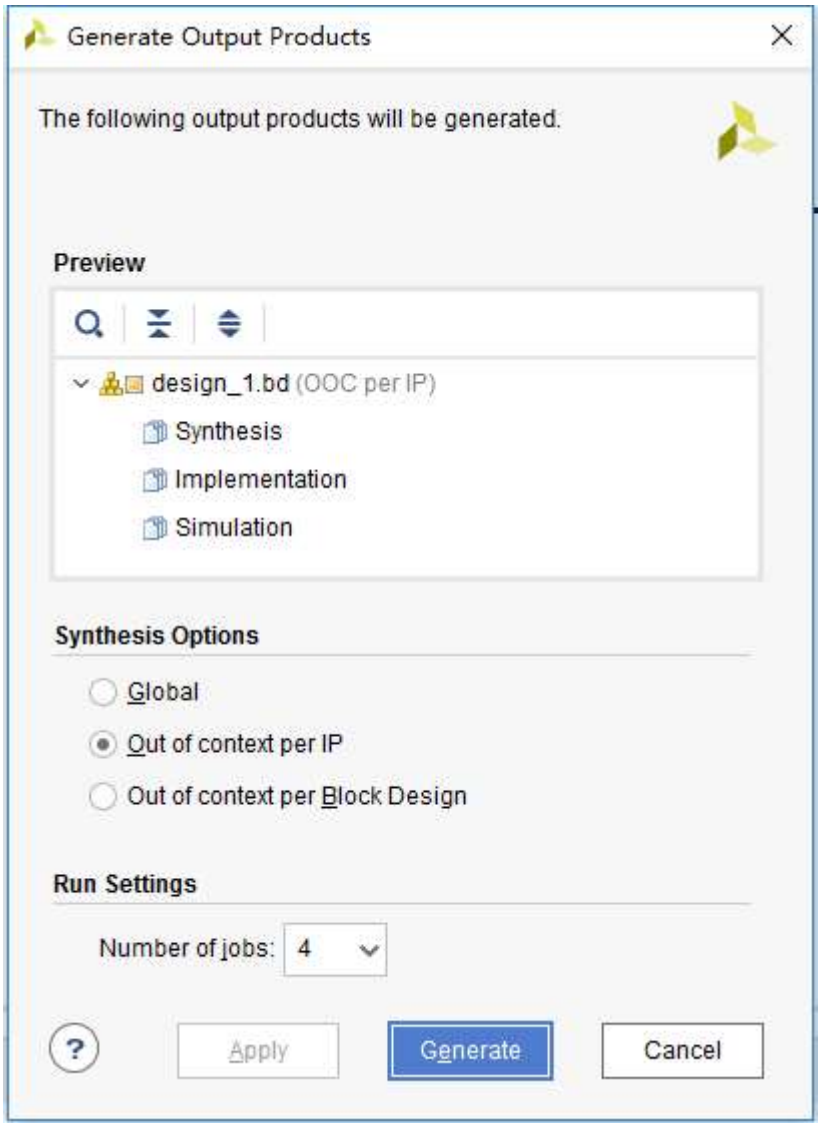
Click **Sources** as shown in the figure below.



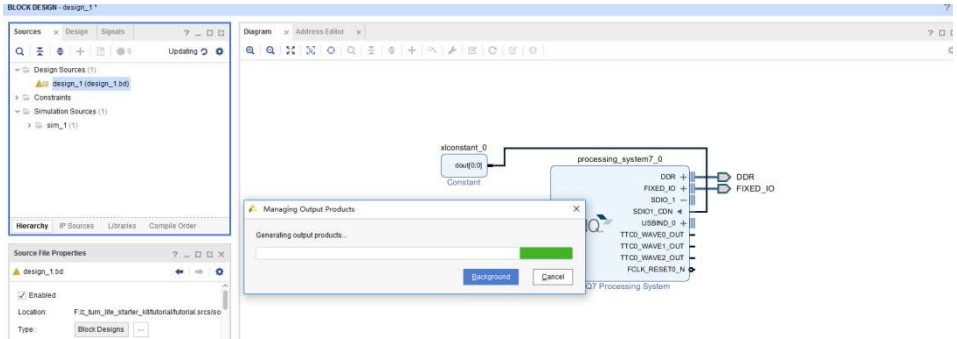
Right-click design_1 and click Generate Output Products



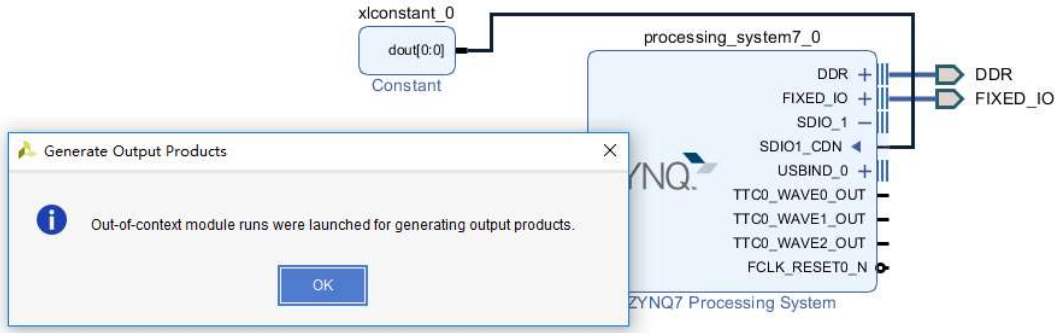
Click Generate



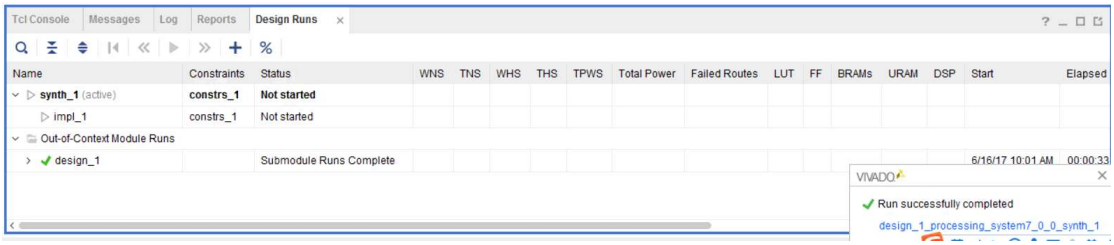
Running



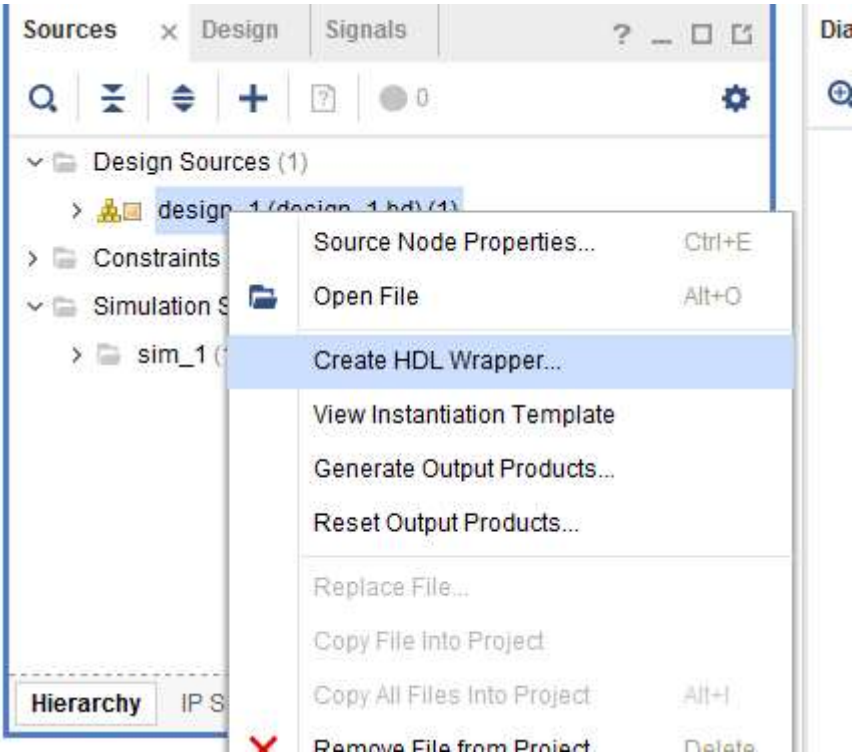
Click **OK**.



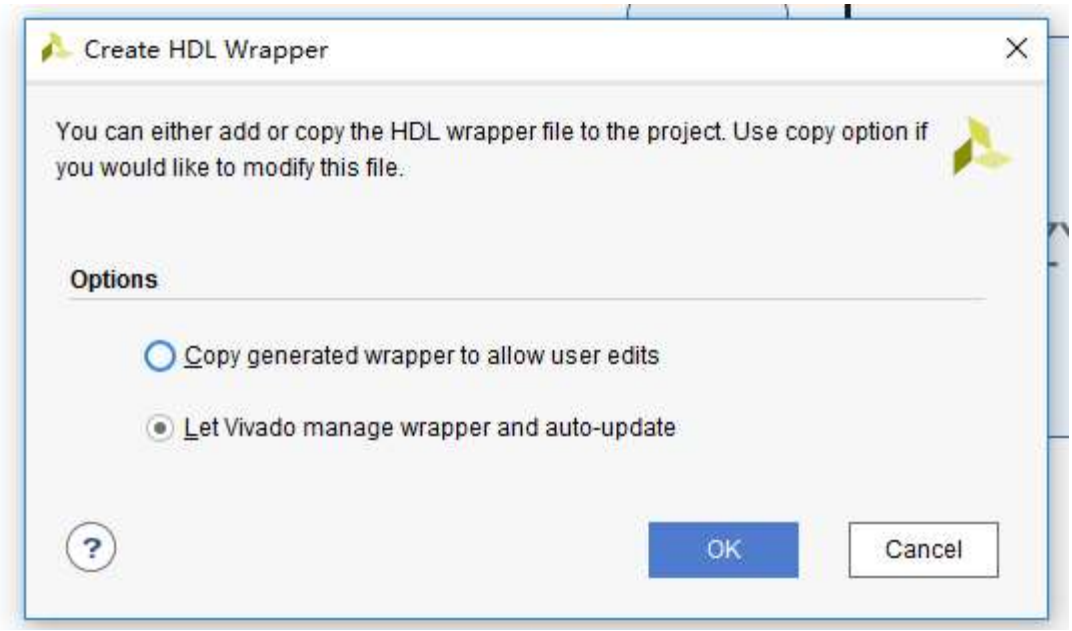
Waiting until Run successfully completed appeared.



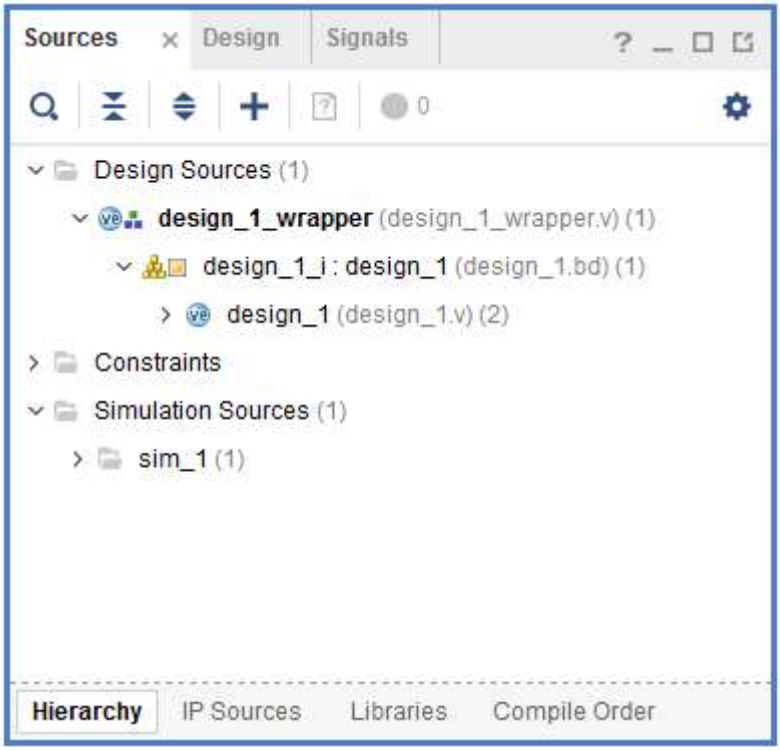
Right-click design_1 and click Create HDL Wrapper



Click **OK**

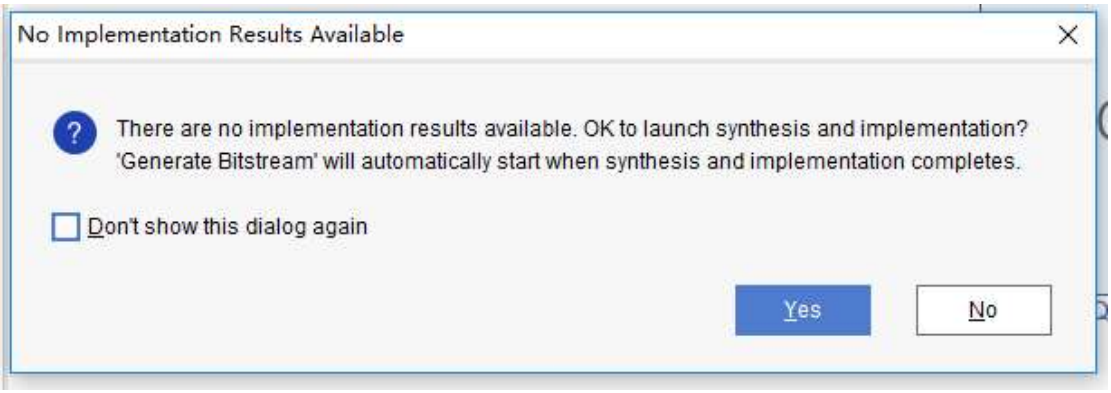


Top module design_1_warpper.v will be Generated automatically.

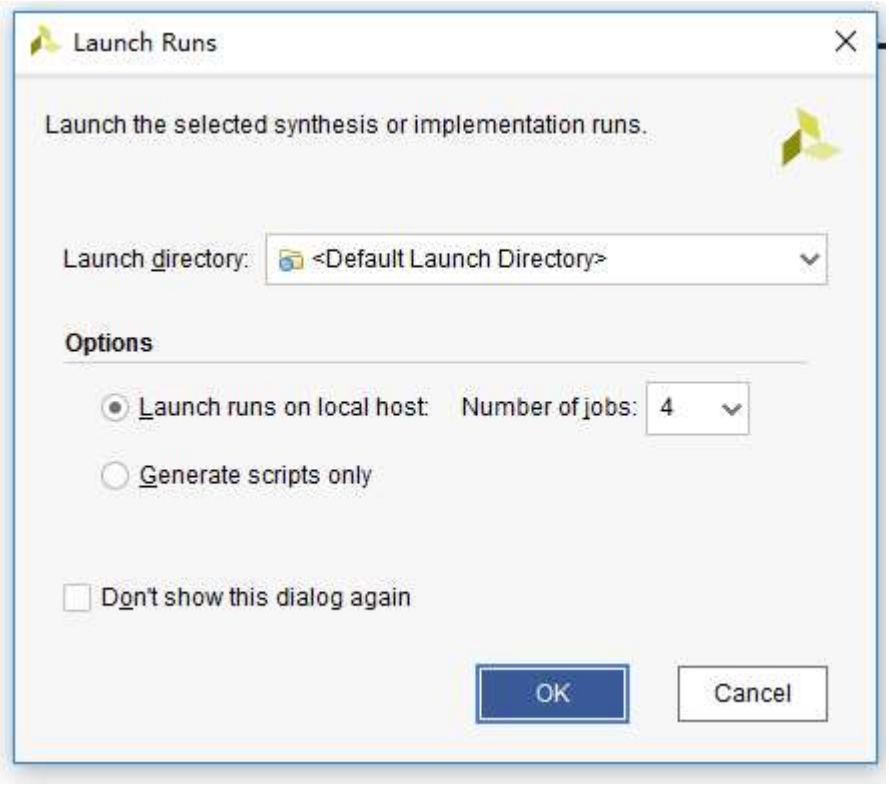


Click **Generating Bitstream** as shown in the figure below.

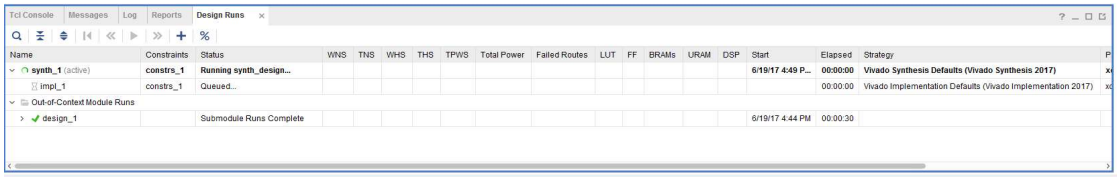
Click **Yes**



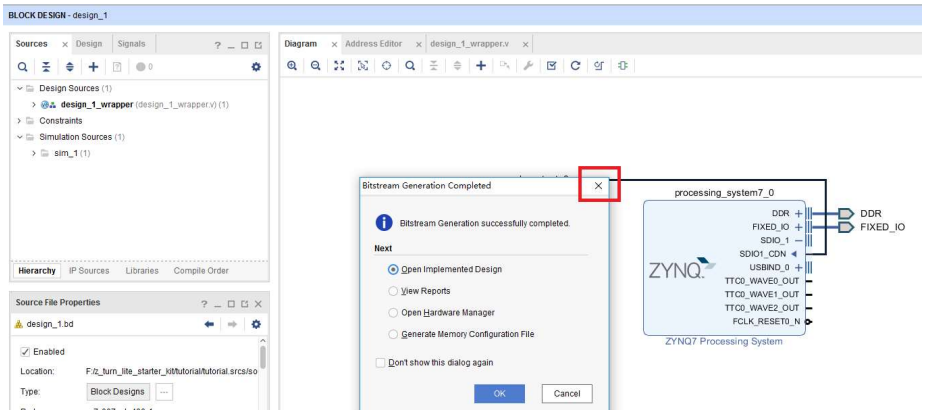
Click **OK**



Running:



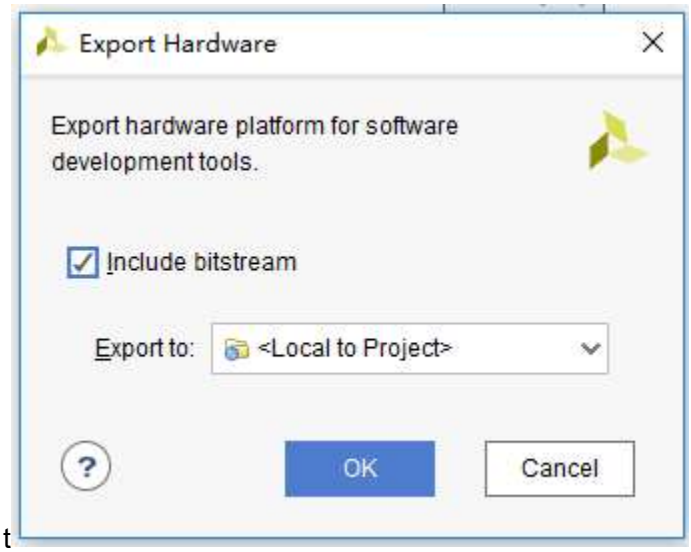
Bitstream Generation Completed, click the dialog appeared



5 Exporting the Hardware Platform to SDK

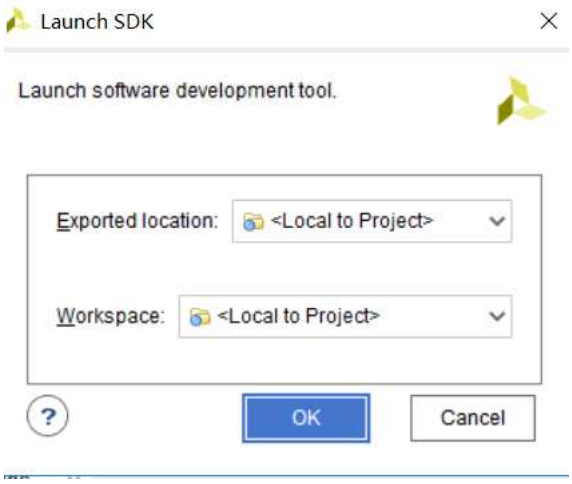
Now that the hardware platform has been generated it must be exported to SDK in order to generate Board Support Package (BSP) as well as all other software applications.

• With the **Implemented Design Open** in Vivado GUI, select **File > Export > Export Hardware** from the Vivado toolbar. When the following dialog box appears, make sure the **Include bitstream** box is checked and click **OK** to continue.

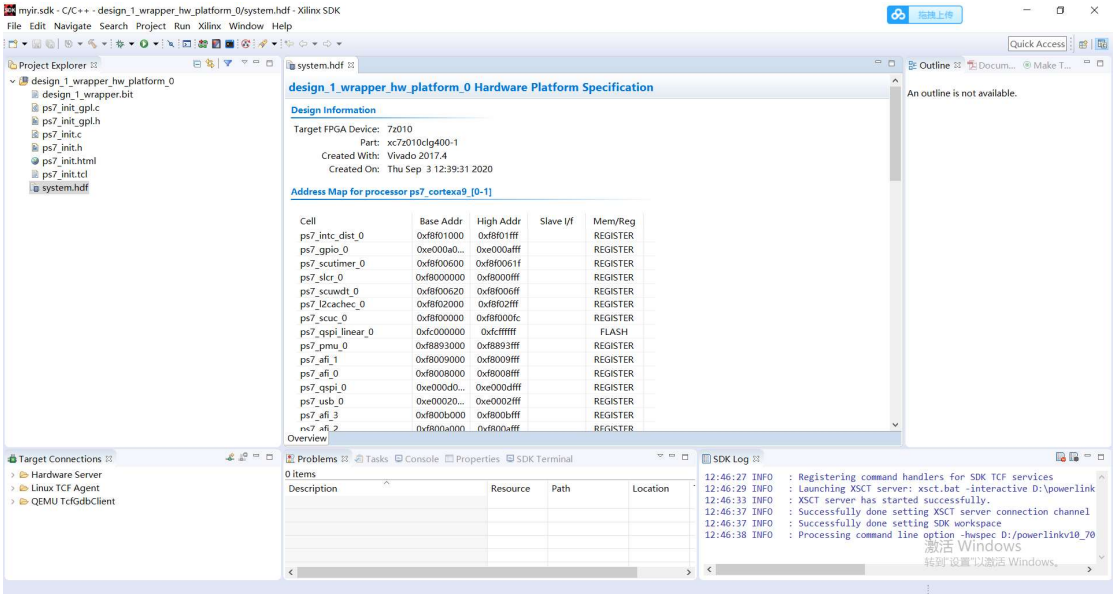


t

• From the Vivado toolbar, select **File > Launch SDK** to open the SDK tool. When the following dialog box appears, click **OK** to continue.



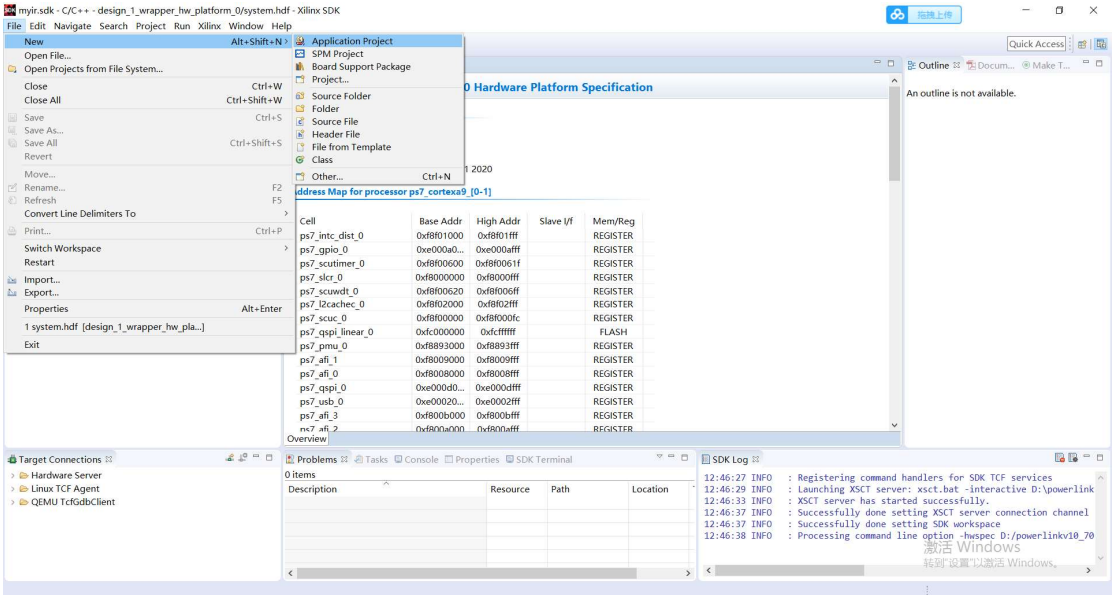
SDK will open, hardware platform will be imported to SDK, and you should see the following in the SDK GUI. The contents of the **design_1_wrapper_hw_platform_0** shown in the following figure willbe used later to generate the BSP for the hardware platform.



5.1Creating a “Hello World” Software Project

we can begin creating software applications directly since the SDK will generate a BSP for us automatically. We will start by creatinga simple “Hello World” application so that you can become familiar with the SDK flow.

• From the SDK toolbar, select **File>New -> Application Project**to Create a new application.

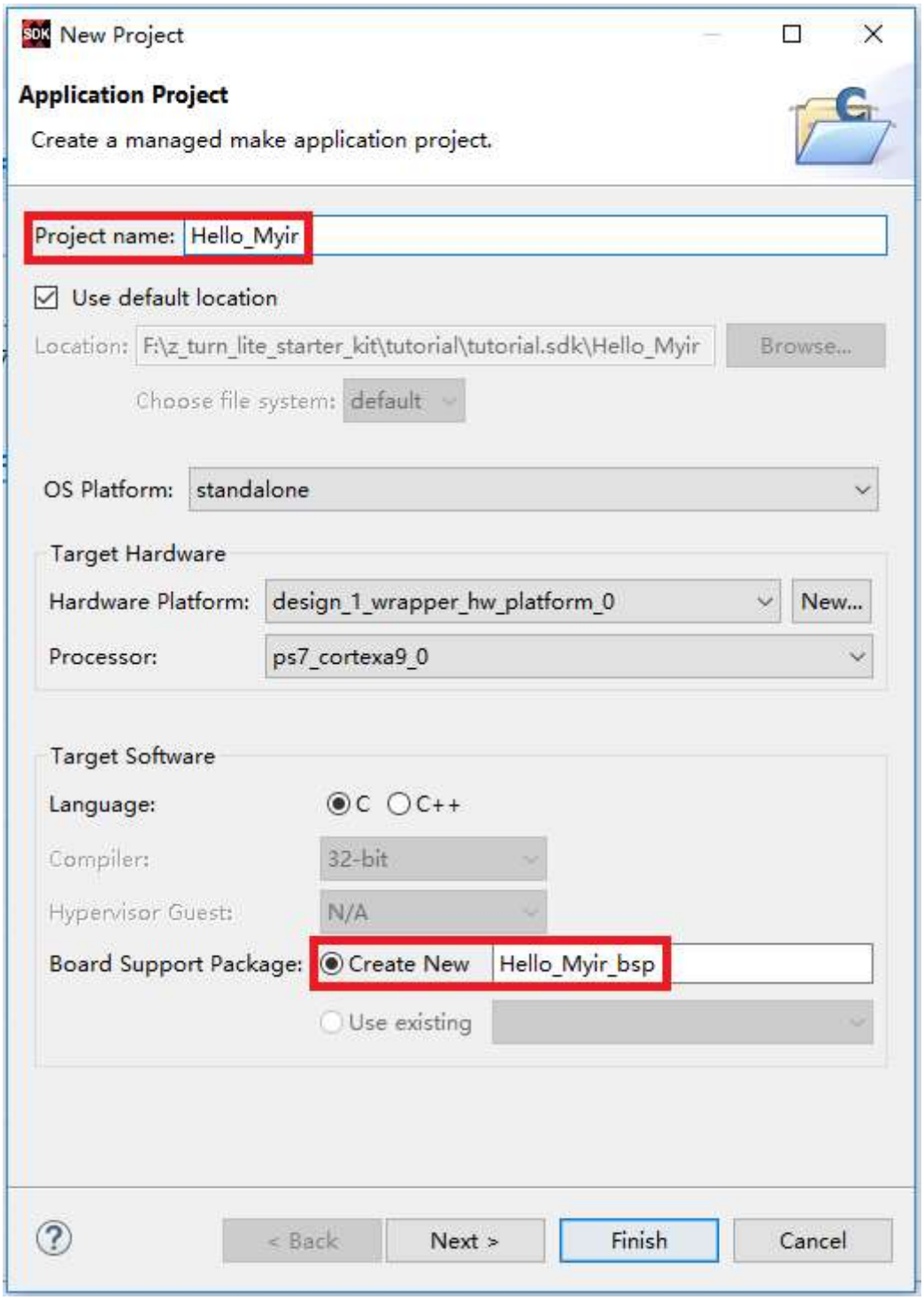


• When the **New Project** dialog box appears:

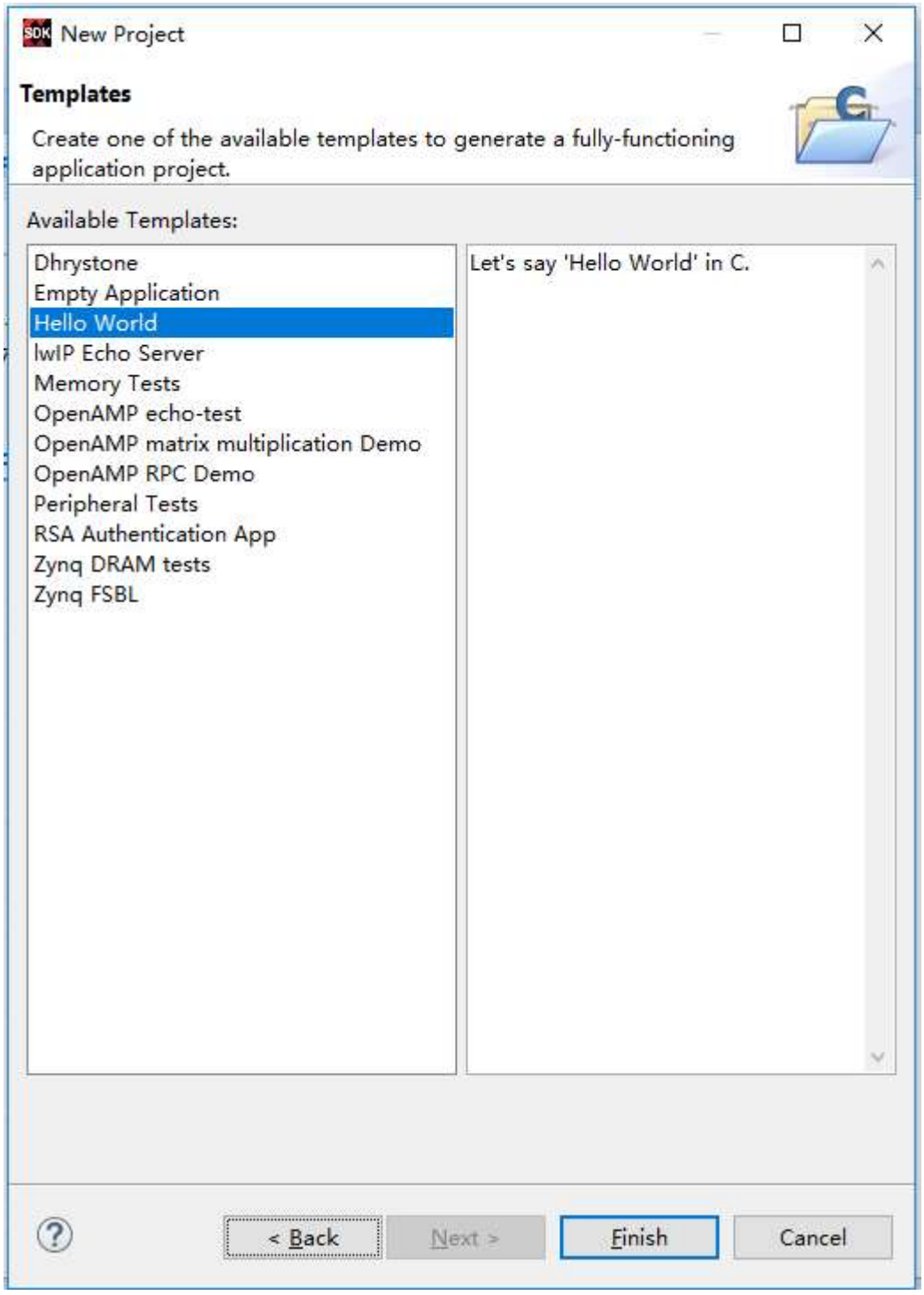
a. Set the Project name to Hello_Myir.

b. Under the **Board Support Package**, click on the **Createnew**radio button.

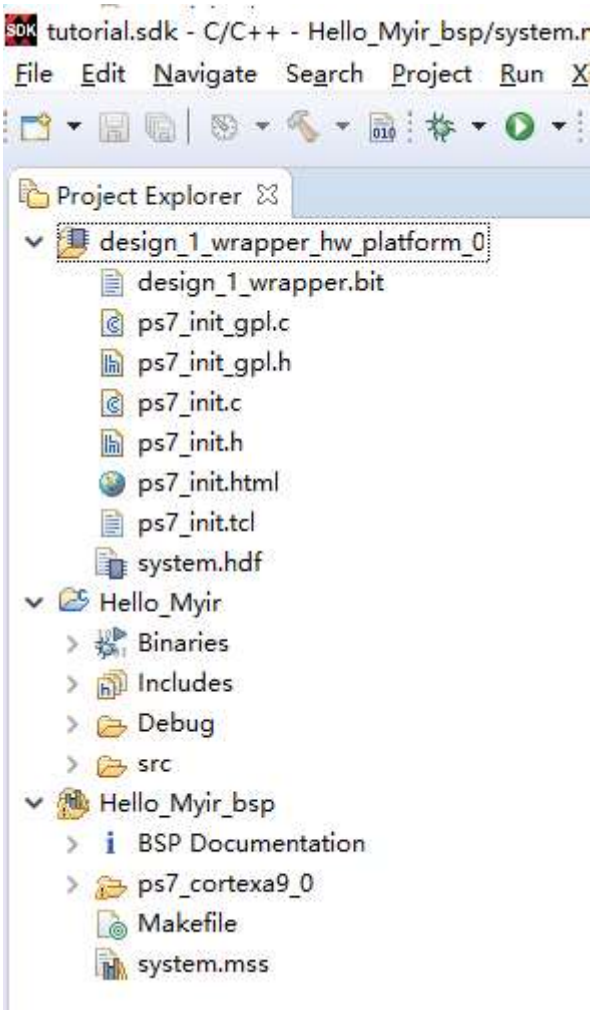
c. Keep all other default options and click on **Next** to continue.



• When the following dialog box appears, click on the **Hello World** under the **Available Templates** and then click on **Finish** to continue.



The **Hello_Myir** software project will be added to the SDK workspace as shown in the following figure.



You will also see the **Hello_Myir** project being compiled in the SDK console window asshown in the following figure. The **Hello_Myir.elf** is the output of the compiler which will beloaded into the MYD-Y7Z020/10/007S DDR3 memory and executed in the next step.

6 Setting up the Hardware

Please perform the following steps to setup the MYD-Y7Z010/007S Starter Kit.

- Turn off switch 1 and turn on switch 2 in the SW1.
- Connect the Xilinx USB Platform Cable DLC9 to the J6 JTAG port and the USB port of the PC. This will provide JTAG connection to the board.
- Connect the MY-UART012U USB to COM cable to the J8 DebugUART port and the USB port of the PC.This will provide USB-UART connection to the board.
- Connect 12V power supply through power cable to J1 on the MYD-Y7Z010/007S.
- Connect an Ethernet cable to the J4 RJ45 connector on the MYD-Y7Z010/007S and the GigabitEthernet port of the PC.
- Start a Putty session and set the serial port parameters to 115200 baud rate, 8 bits, 1stop bit, no parity and no flow control (please refer to the **Setting up the Host PC** section atthe end of this document for installing the software driver for the USB-UART port and settingup the UART).
- Set the IP address of your PC to **192.168.1.1** with subnet mask of **255.255.255.0**.

7 Running the Hello World Program on Hardware

In this section, we will be using the JTAG debug interface to load the **hello_myir.elf** executable file into the MYD-Y7Z020/10/007S DDR3 memory and run it.

• From the SDK GUI **Project Explorer**, right click on the **hello_myir** software project and then select **Run As > 1 Launch on Hardware (System Debugger)** as shown in the following figure. This command will load the **hello_myir.elf** executable file into the MYD-Y7Z020/10/007S DDR3 memory and run it on the board.

You should see the Hello World being displayed on the putty terminal as show below.

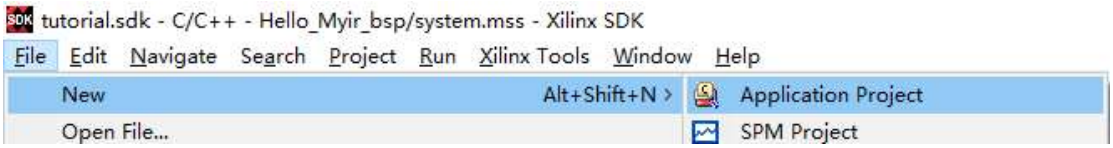
8 Generating Application Software

Now that a simple Hello World program has been loaded into the DDR3 memory and executed, in the next sections will try loading other software applications into the DDR3 memory and run them on the board.

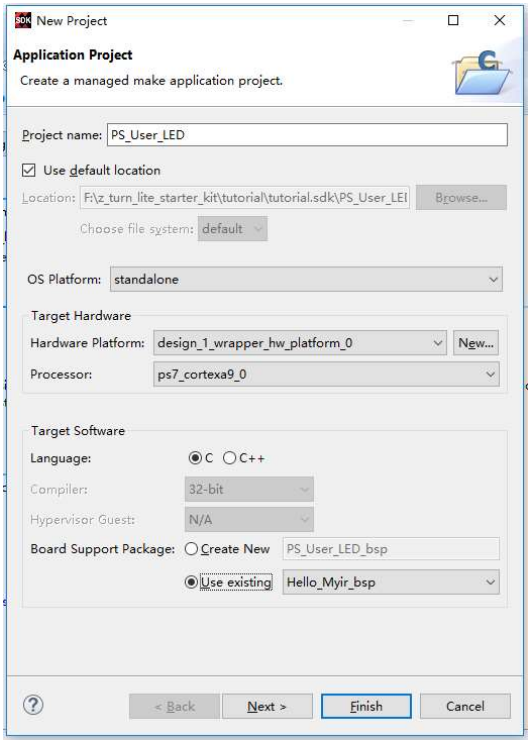
8.1 Flashing the PS User LED

In this section we will use a simple software application (already developed for you) to flash the PS user LED on the MYD-Y7Z020/10/007S Starter Kit. The first step is to create a software project in SDK for the PS user LED.

• From the SDK toolbar, select **File > New > Application Project** as shown in the following figure.



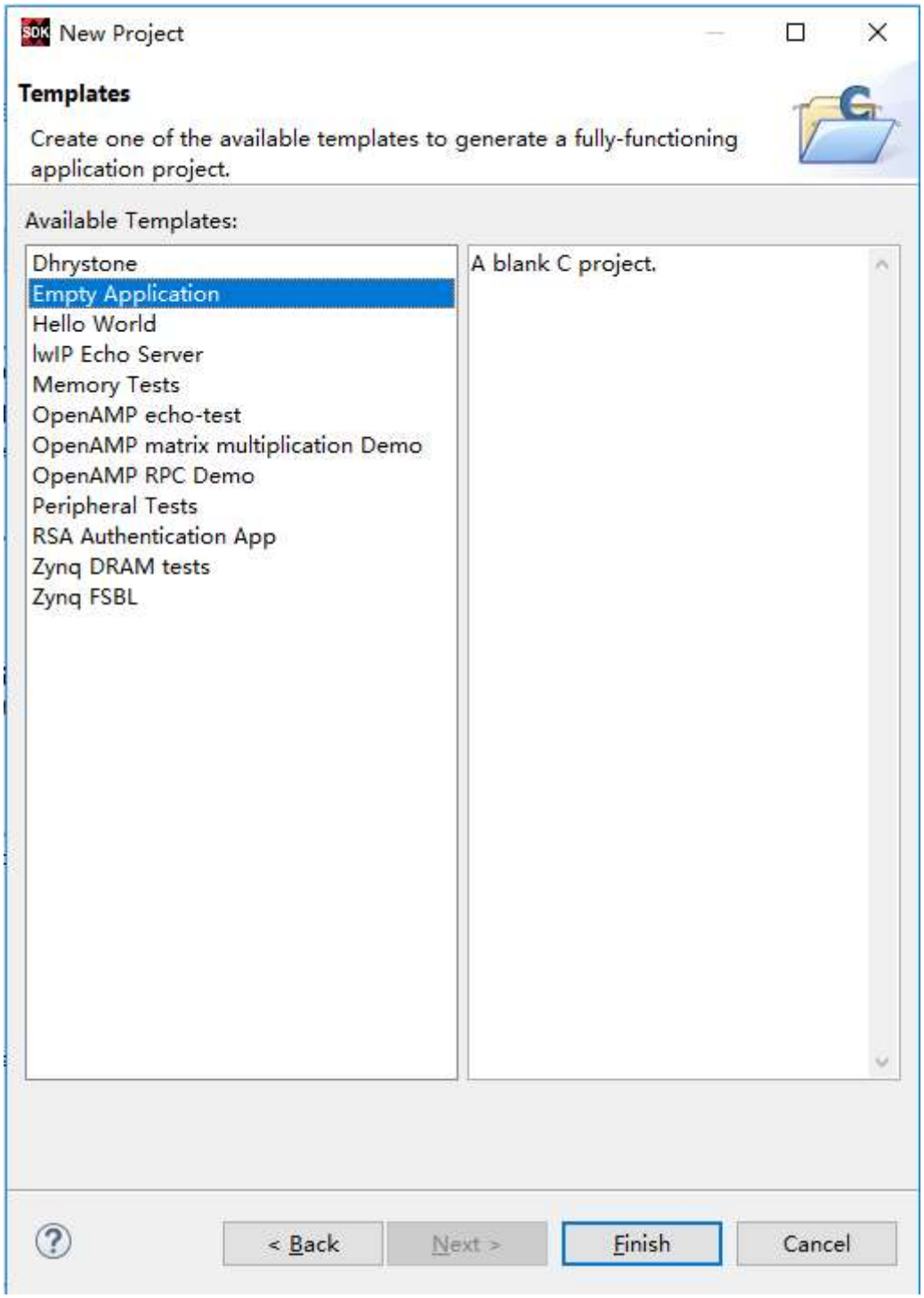
• When the **New Project** dialog box appears:
a. Set the Project name to PS_User_LED.
b. Under the **Board Support Package**, click on the **Use existing** radio button and select **Hello_Myir_bsp** from the drop-down menu.
c. Keep all other default options and click on **Next** to continue.



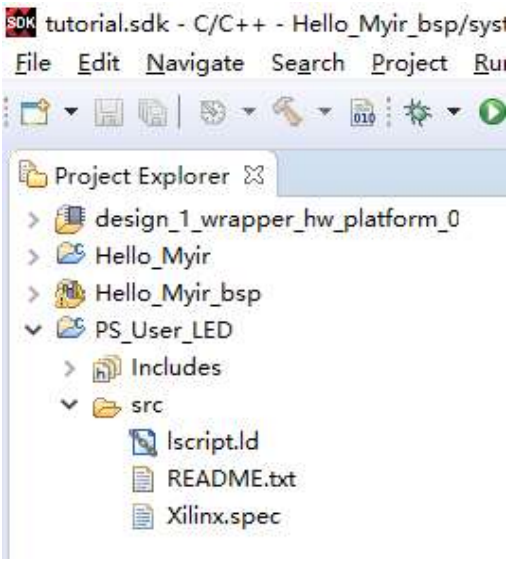
• When the following dialog box appears:

a. Click on the **Empty Application**. This will create the **PS_User_LED** software project without any source code. We will add the source code to the project in the next step.

b. Click on **Finish** to continue.

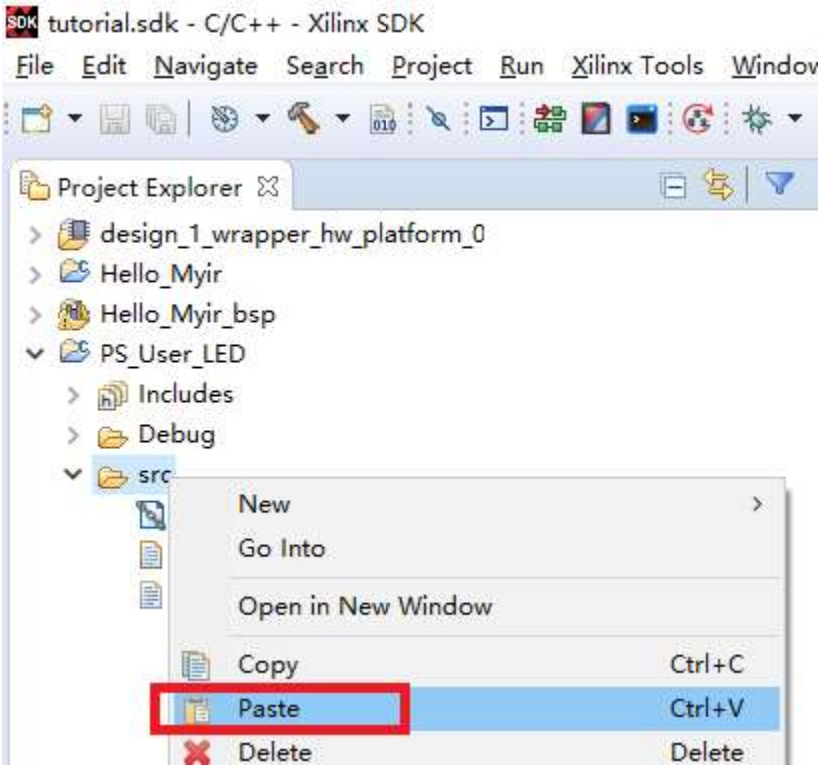


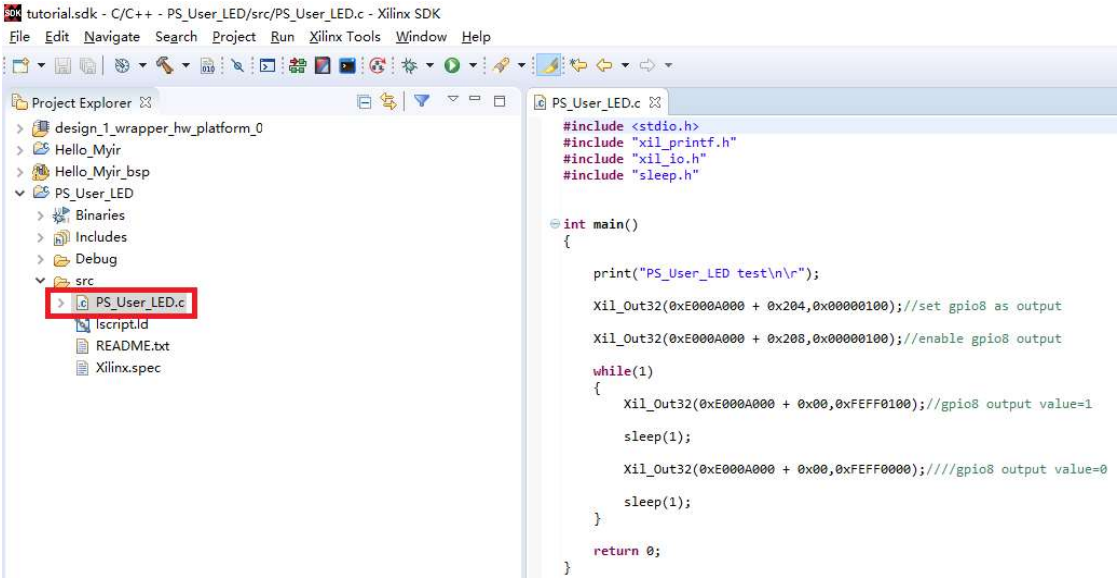
The SDK Project Explorer should look as shown in the following figure (you would need to expand the **PS_User_LED** software project to see the **src** folder. As can see, there is no source code under the **src** folder for this project. We will add the source code in the next step.



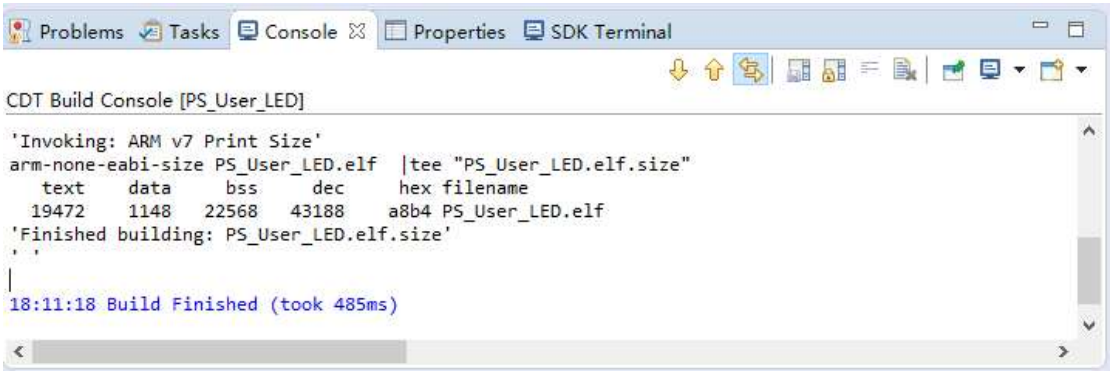
Copy PS_User_LED.c from iso disk.
(PATH: *\05-Programmable_Logic\7z010\ src)

Paste it to **src** on **PS_User_LED** project.

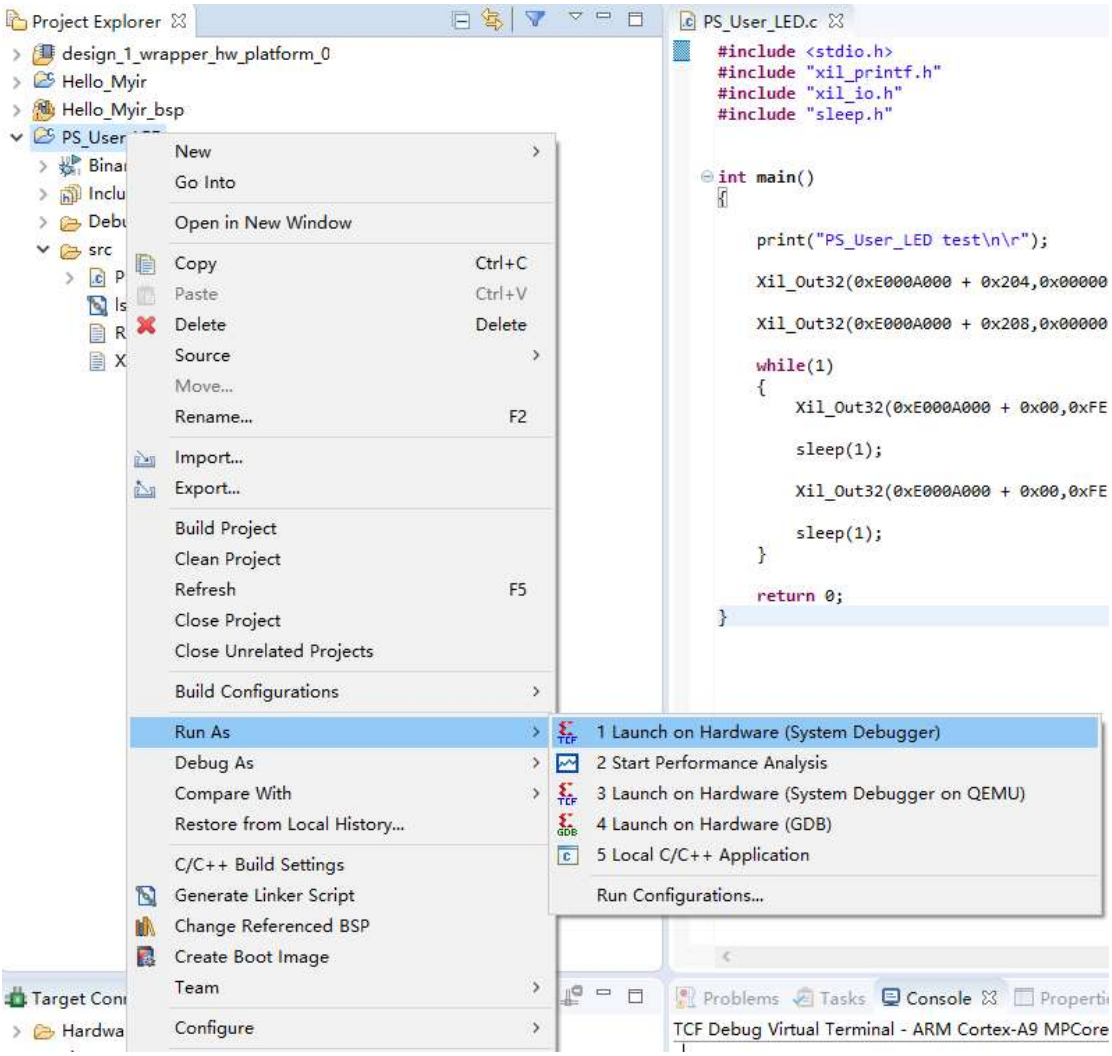




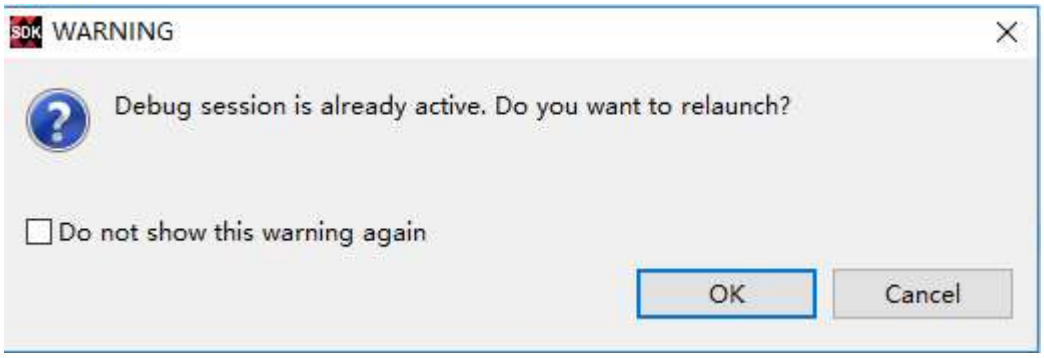
You will also see the **PS_User_LED** project being compiled in the SDK console window as shown in the following figure. The **PS_User_LED.elf** is the output of the compiler which will be loaded into the MYD-Y7Z020/10/007S DDR3 memory and executed in the next step.



• From the SDK GUI **Project Explorer**, right click on the **PS_User_LED** software project and then select Run As > 1 Launch on Hardware (System Debugger) as shown in the following figure. The **PS_User_LED.elf** executable file will be loaded into the DDR3 memory and run on the board.



• When the following dialog box appears, click **OK** to continue.



You should see the **PS User LED (D60)** flashing on the starter kit.

8.2 Testing the Gigabit Ethernet Port

In this section we will create an SDK software project to verify the operation of the Gigabit Ethernetport on the MYD-Y7Z020/10/007S board.

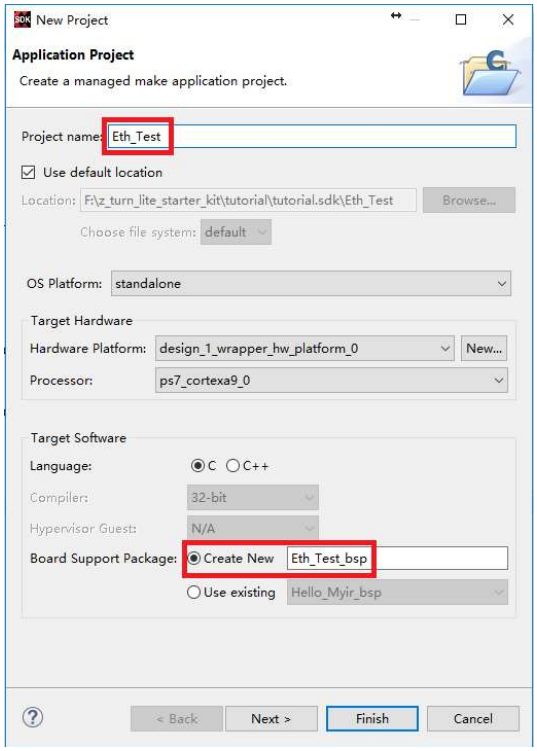
• From the SDK toolbar, select **File > New > Application Project**.

• When the **New Project** dialog box appears:

a. Set the Project name to Eth_Test.

b. Under the **Board Support Package**, keep the default **Create New** with **Eth_Test_bsp** as the name.

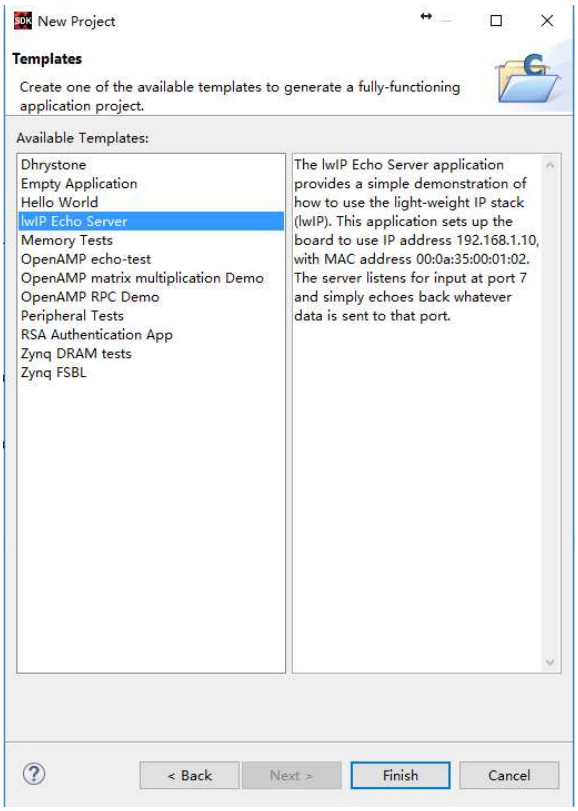
c. Keep all other default options and click on **Next** to continue.



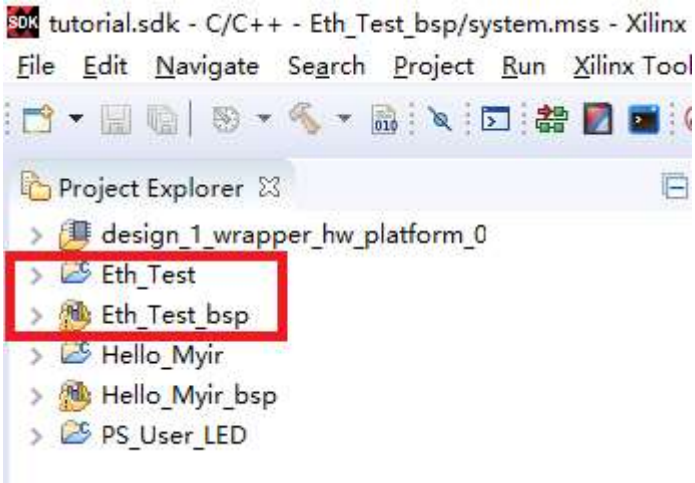
• When the following dialog box appears:

a. Click on the **lwIP Echo Server** to highlight it.

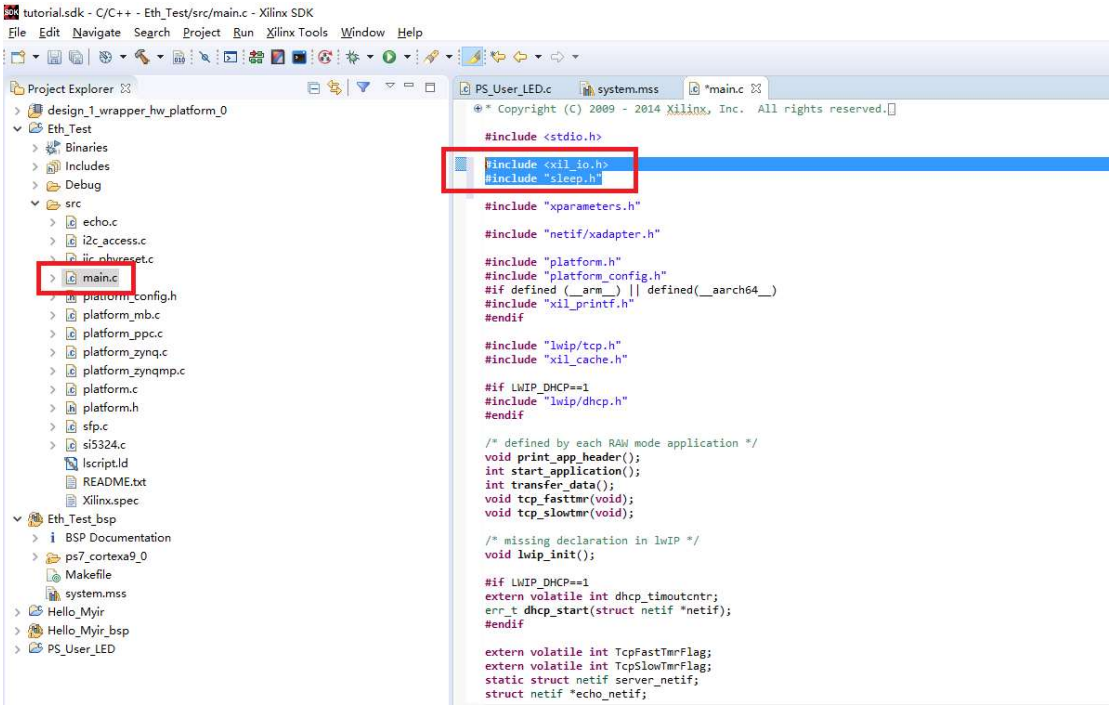
b. Click on **Finish** to continue.



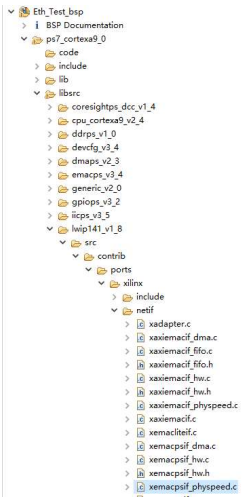
The **Eth_Test** software project and the **Eth_Test_bsp** BSP file will be generated.



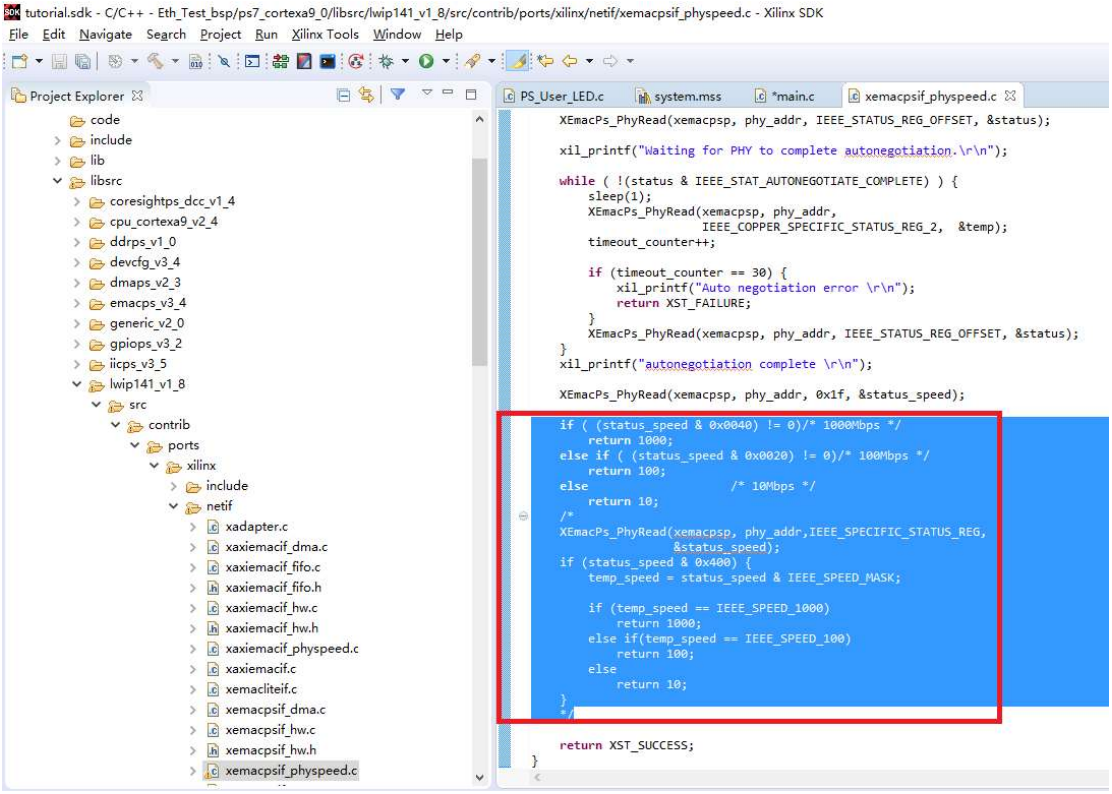
Add code to main.c as shown in the figure belowand type **Ctrl+S** to save.
(PATH: *\05-Programmable_Logic\7z010\ src\Eth_Test\)



Modify Eth_Test_bsp->ps7_cortexa9_0->libsrc\lwip141_v1_8->src->contrib->ports->**xilinx->netif->xemacpsif_physpeed.c** as shown in the figure below and type **Ctrl+S** to save.The code will beautomatically compiled, and the **Eth_Test.elf**executable file will be automatically generated as shown in the following figure. In the next section we will load the **Eth_Test.elf** executable file into the DDR3 memory and run it on the board.

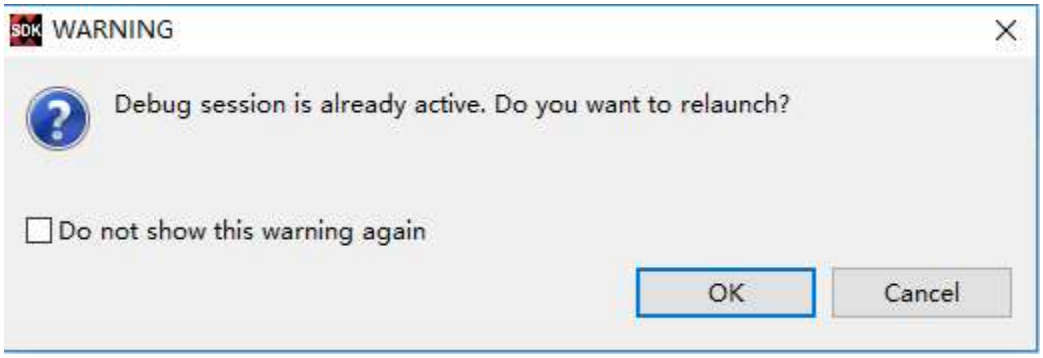


The modification below is located in the end of function
static u32_t get_Marvell_phy_speed(XEmacPs *xemacpsp, u32_t phy_addr)
(You can refer to the **xemacpsif_physpeed.c** file in the iso disk we offered
Path: *\05-Programmable_Logic\7z010\src\Eth_Test)

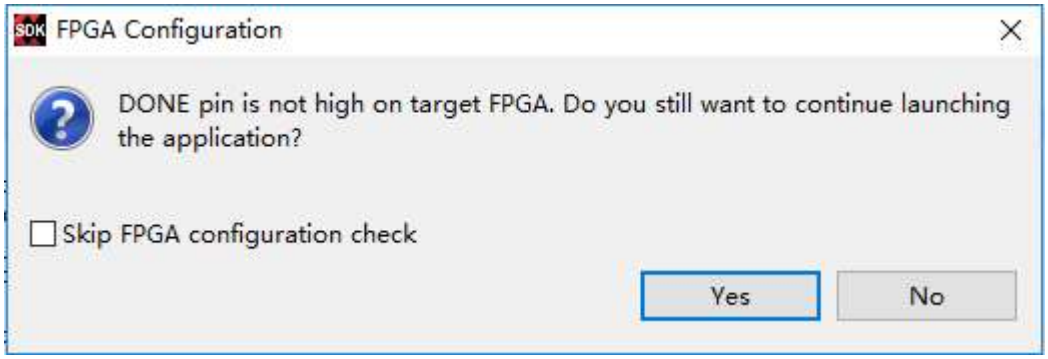


• From the SDK GUI **Project Explorer**, right click on the **Eth_Test** software project and then select **Run As > 1 Launch on Hardware (System Debugger)** as shown in the following figure. The **Eth_Test.elf** executable file will be loaded into the DDR3 memory and run on the board.

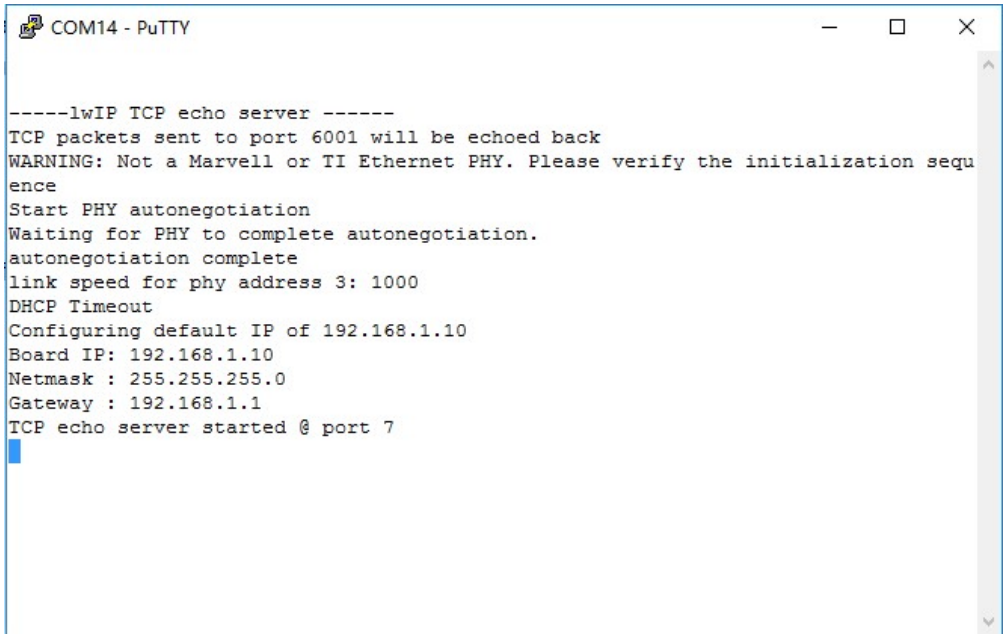
• When the following dialog box appears, click **OK** to continue.



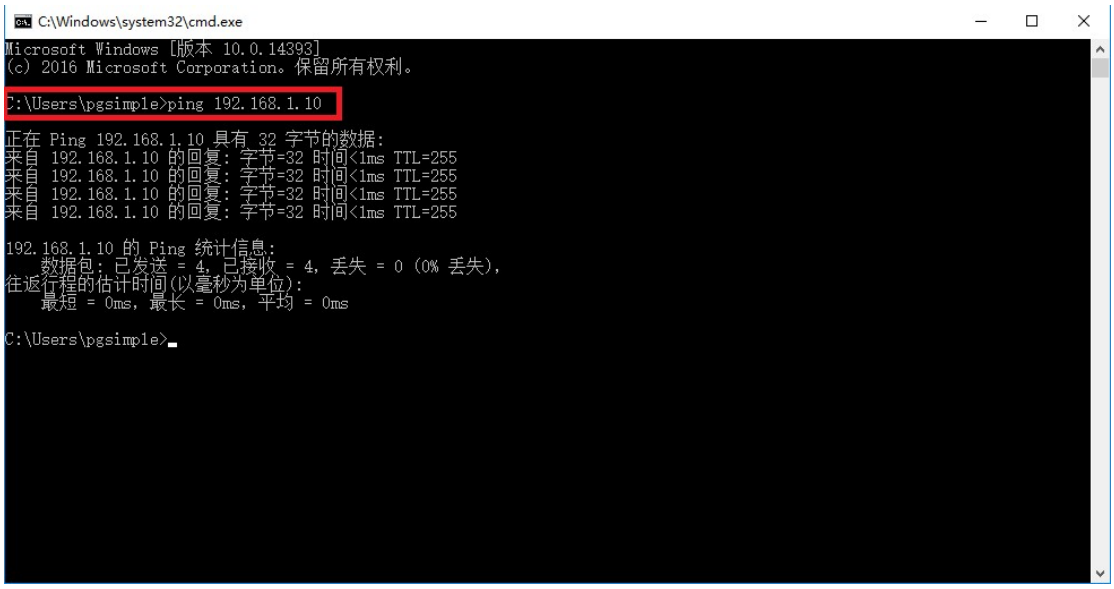
• When the following dialog box appears, click **Yes** to continue.



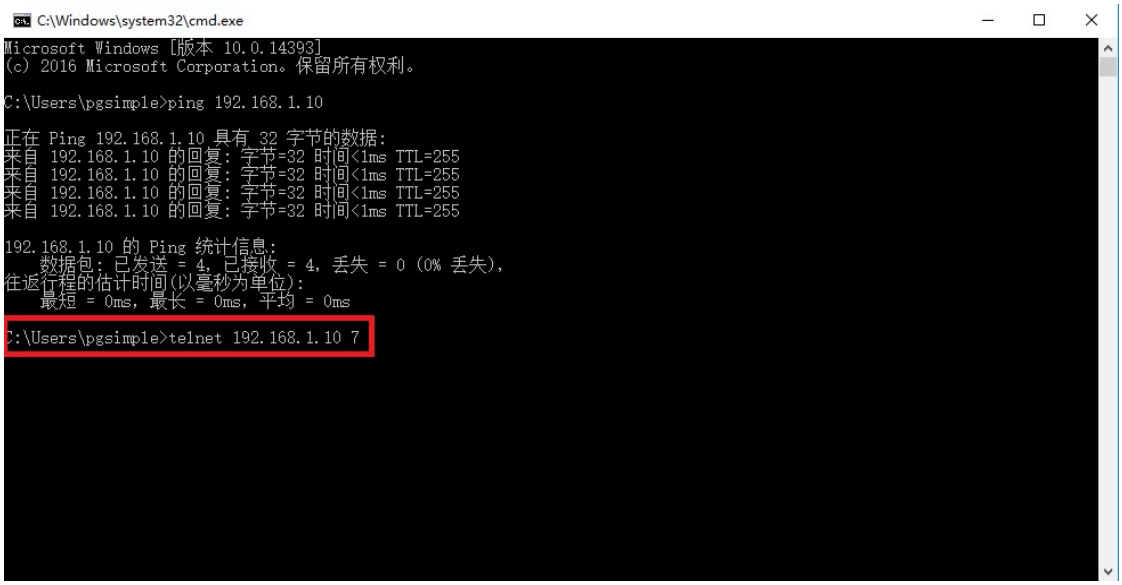
• Wait until the lwIP Echo Server output is displayed on the Tera Term terminal as shown in the following figure. At this time, the Ethernet port has gone through the auto-negotiation and is ready for normal operation.



• Open a command window and ping the board using the board IP address of **192.168.1.10** as shown below. You should see 4 replies back as shown.



• To connect to the echo server, use the telnet utility program. Type the following telnet command as shown below and hit the return key.



If the echo server works properly, any data sent to the board is echoed in response. Some telnet clients immediately send the character to the server and echo the received data back instead ofwaiting for the carriage return. Simply type a few characters and see them echoed back on the terminal.

9 Booting From Primary/Secondary Boot Devices

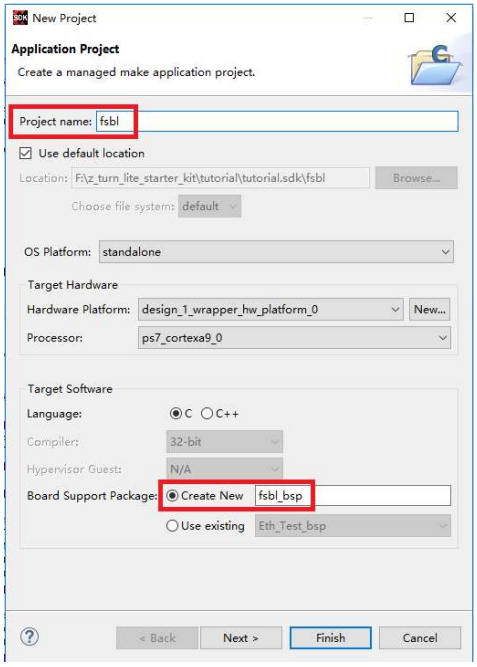
In the previous sections, the SDK debugger was used to load user executables into the MYD-Y7Z020/10/007S DDR3 memory over the JTAG interface and run the programs on the board. This method of operation is ideal for debug environment. However, in normal operation user code and bitstream areloaded into the MYD-Y7Z020/10/007S DDR3.

In the following sections we will show you how to:
- Generate boot image for each boot device
- Program the image into the boot device
- Boot from SD Card boot device

9.1Generating the First Stage Boot Loader (FSBL)

In order to boot from any of the boot devices, a bootloader is required to load the user executable and bitstream into the MYD-Y7Z020/10/007S DDR3 and the ZynqSoC. When the Zynq initially boots, its internal ROM loads the bootloader (FSBL) from the external boot deviceinto the ZynqSOC internal SRAM and releases the control to the FSBL.FSBL will then load the user executable from the external boot device into the MYD-Y7Z020/10/007S DDR3, loads the bitstream from the external boot device into the ZynqSOC PL, and then releases the control to the user executable. The following steps describe how to generate the FSBL.
- From the SDK toolbar, select **File > New > Application Project**.
- When the **New Project** dialog box appears:

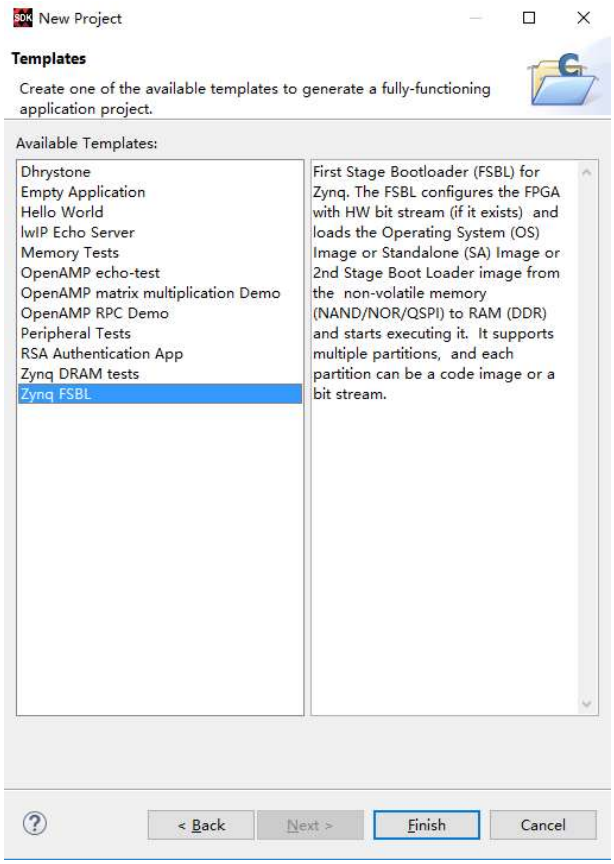
a. Set the Project name to fsbl.
b. Under the **Board Support Package**, keep the default **Create New** with **fsbl_bsp** as the name.
c. Keep all other default options and click on **Next** to continue.



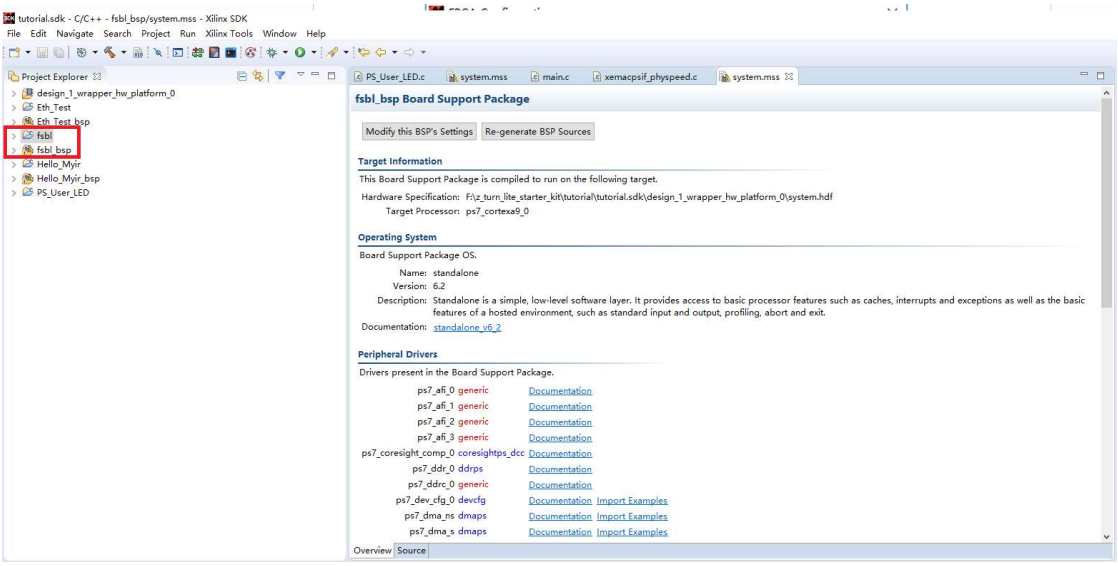
• When the following dialog box appears:

a. Click on the **Zynq FSBL** to highlight it.

b. Click on **Finish** to continue.



The **fsbl** software project and the **fsbl_bsp** BSP file will be generated, the code will be automatically compiled, and the **fsbl.elf** executable file will be automatically generated as shown in the following figure.

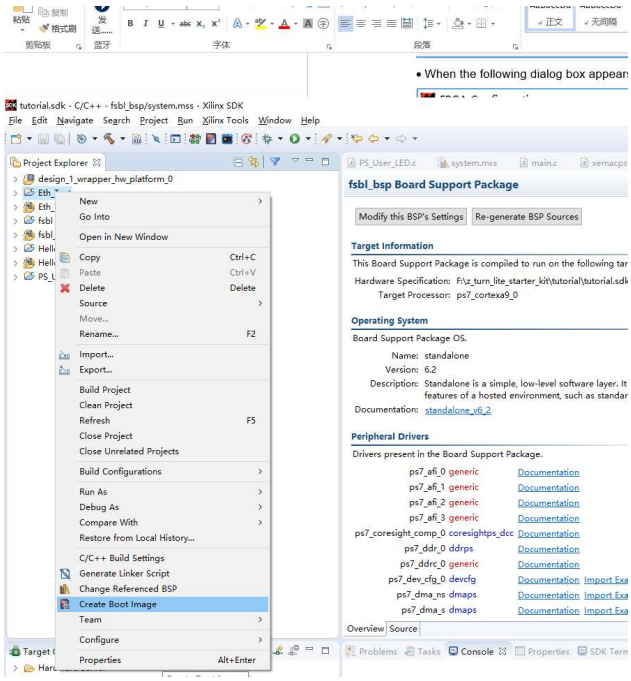

tutorial.sdk - C/C++ - fsbl_bsp/system.mss - Xilinx SDK
File Edit Navigate Search Project Run Xilinx Tools Window Help
Project Explorer
design_1_wrapper_hw_platform_0
Eth_Test
Eth_Test_bsp
fsbl
fsbl_bsp
Hello_Myir
Hello_Myir_bsp
PS_User_LED
PS_User_LED.c
system.mss
main.c
xemacpsif_physpeed.c
system.mss
fsbl_bsp Board Support Package
Modify this BSP's Settings
Re-generate BSP Sources
Target Information
This Board Support Package is compiled to run on the following target.
Hardware Specification: F:\z_turn_lite_starter_kit\tutorial\tutorial.sdk\design_1_wrapper_hw_platform_0\system.hdf
Target Processor: ps7_cortexa9_0
Operating System
Board Support Package OS.
Name: standalone
Version: 6.2
Description: Standalone is a simple, low-level software layer. It provides access to basic processor features such as caches, interrupts and exceptions as well as the basic features of a hosted environment, such as standard input and output, profiling, abort and exit.
Documentation: standalone_v6_2
Peripheral Drivers
Drivers present in the Board Support Package.
ps7_afi_0 generic Documentation
ps7_afi_1 generic Documentation
ps7_afi_2 generic Documentation
ps7_afi_3 generic Documentation
ps7_coresight_comp_0 coresightps_dcc Documentation
ps7_ddr_0 ddrps Documentation
ps7_ddrc_0 generic Documentation
ps7_dev_cfg_0 devcfg Documentation Import Examples
ps7_dma_ns dmaps Documentation Import Examples
ps7_dma_s dmaps Documentation Import Examples
Overview Source

9.2Booting from the microSD Card

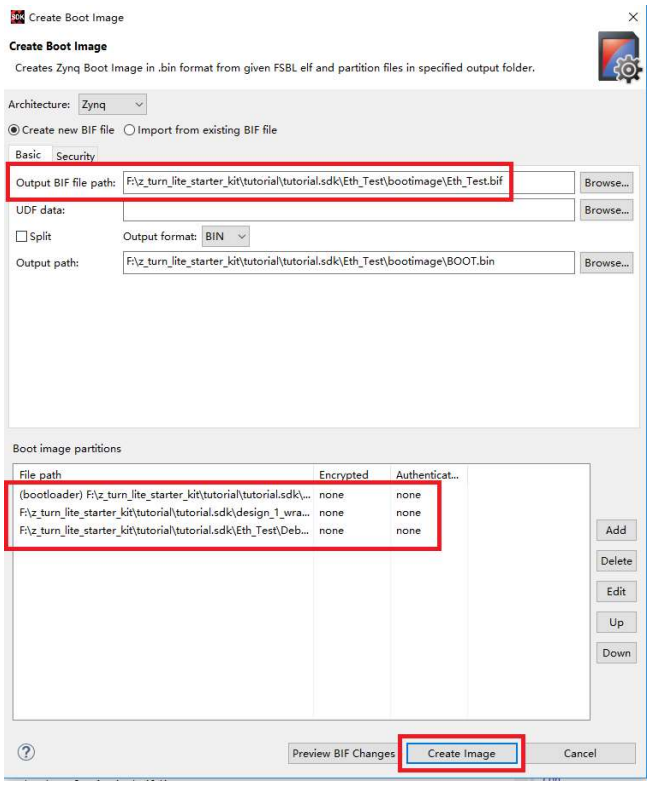
In order to boot from the microSD card, the boot image must be generated and stored on the microSD card.

9.2.1 Generating the SD Card Boot Image

• From the SDK GUI, ritht-click Eth_Test project and select **Create Boot Image** as shown in the following figure.



Click Create Image



The **BOOT.bin** and **Eth_Test.bif** files will be generated in the **Eth_test/bootimage**folder as shown in the following figure.

9.2.2 Copying the SD Card Boot Image onto the microSD Card

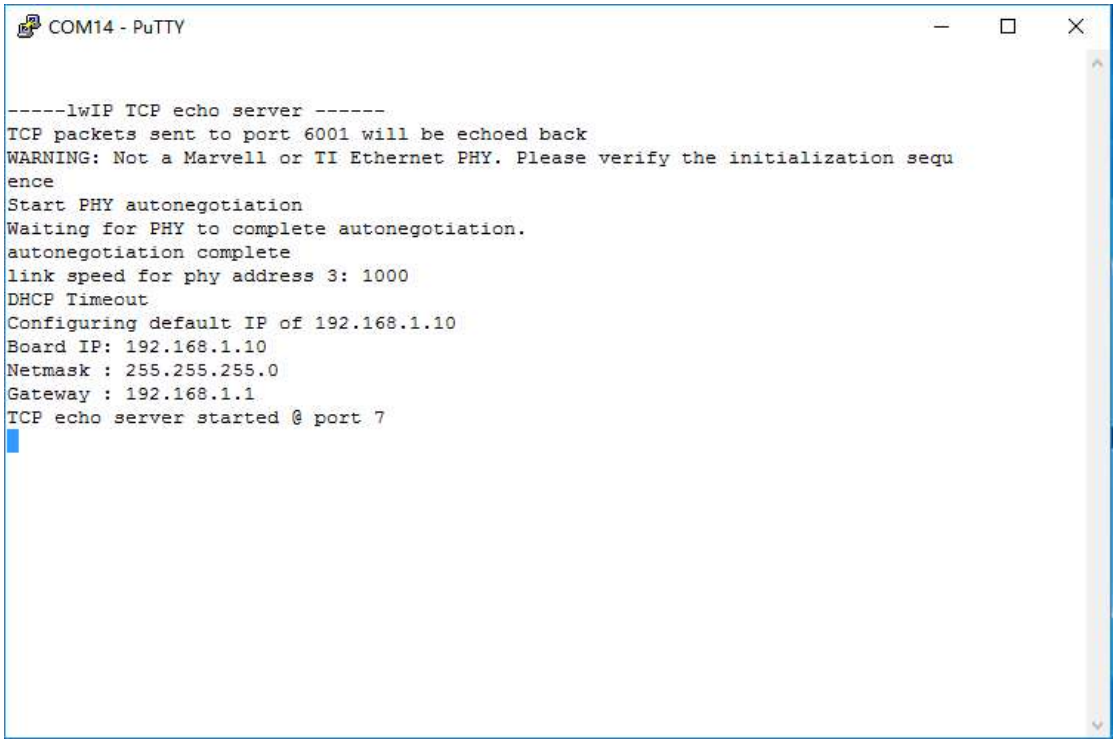
The MYIR MYD-Y7Z020/10/007S Starter Kit is shipped with an 4GB microSD card. Using your PC's memorycard reader slot, copy the **BOOT.bin** file generated in the previous step onto the root directly of the microSD card (you can also use a USB memory card reader adapter, if your PC does not have a microSD card reader slot). Remove the microSD card from the card reader.

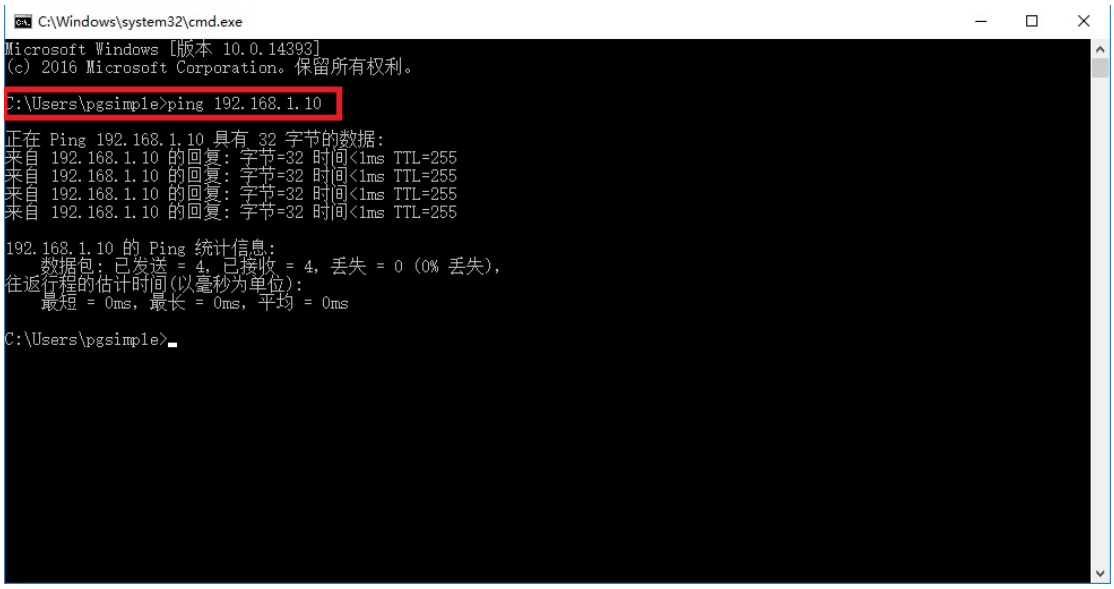
Note: Please make sure to save the original contents of the MYIR provided microSD card on your PC hard drive prior to executing the above **BOOT.bin** copy task.

9.2.3 Booting the Board Using the microSD Card

- Turn off switch 1 and turn on switch 2 in the SW1.
- Connect 12V power supply through power cable to J1 on the MYD-Y7Z020/10/007S.

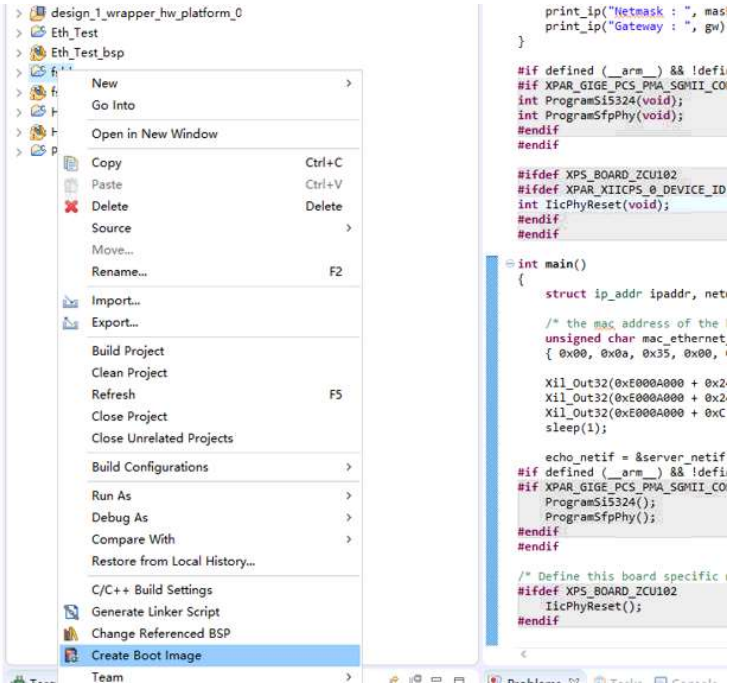
The board will boot, the D7 on the MYD-Y7Z020/10/007S will be illuminated indicating the PL has been configured, and you will see the following on the putty terminal (you may need to wait a few seconds for the complete Echo Server output to be displayed on the terminal).



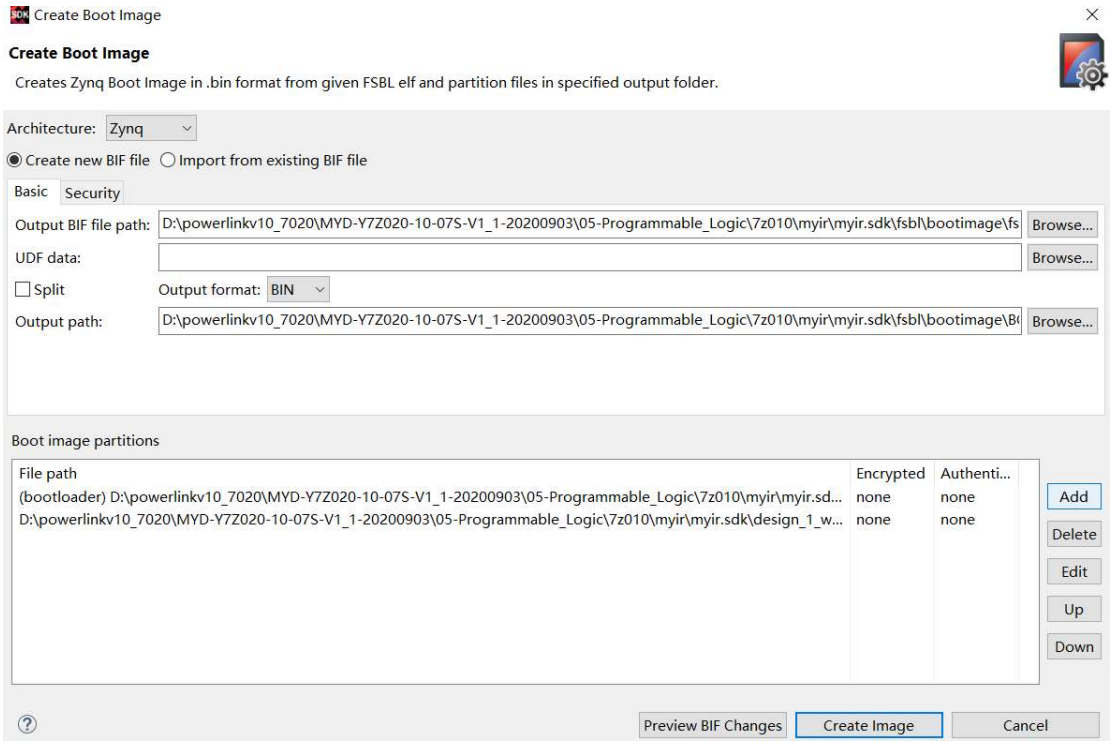

C:\Windows\system32\cmd.exe
Microsoft Windows [版本 10.0.14393]
(c) 2016 Microsoft Corporation。保留所有权利。
C:\Users\pgsimple>ping 192.168.1.10
正在 Ping 192.168.1.10 具有 32 字节的数据:
来自 192.168.1.10 的回复: 字节=32 时间<1ms TTL=255
来自 192.168.1.10 的回复: 字节=32 时间<1ms TTL=255
来自 192.168.1.10 的回复: 字节=32 时间<1ms TTL=255
来自 192.168.1.10 的回复: 字节=32 时间<1ms TTL=255
192.168.1.10 的 Ping 统计信息:
数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
最短 = 0ms, 最长 = 0ms, 平均 = 0ms
C:\Users\pgsimple>

10 Generating linuxboot.bin

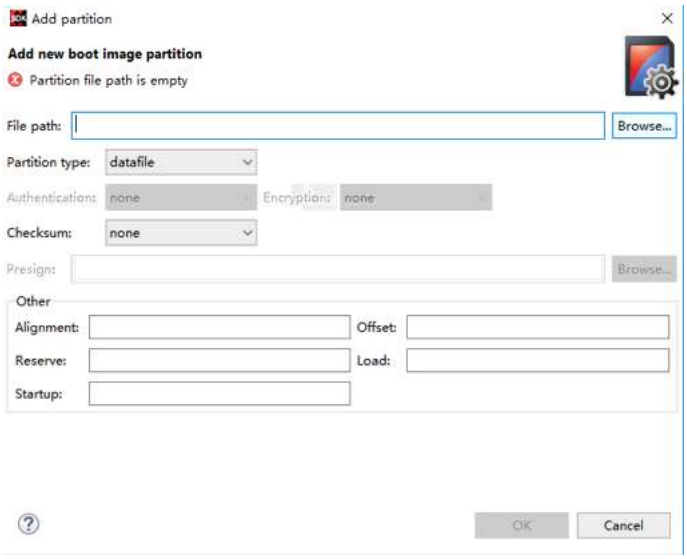
•In SDK manager, Right-click fsbl and click Create Boot Image



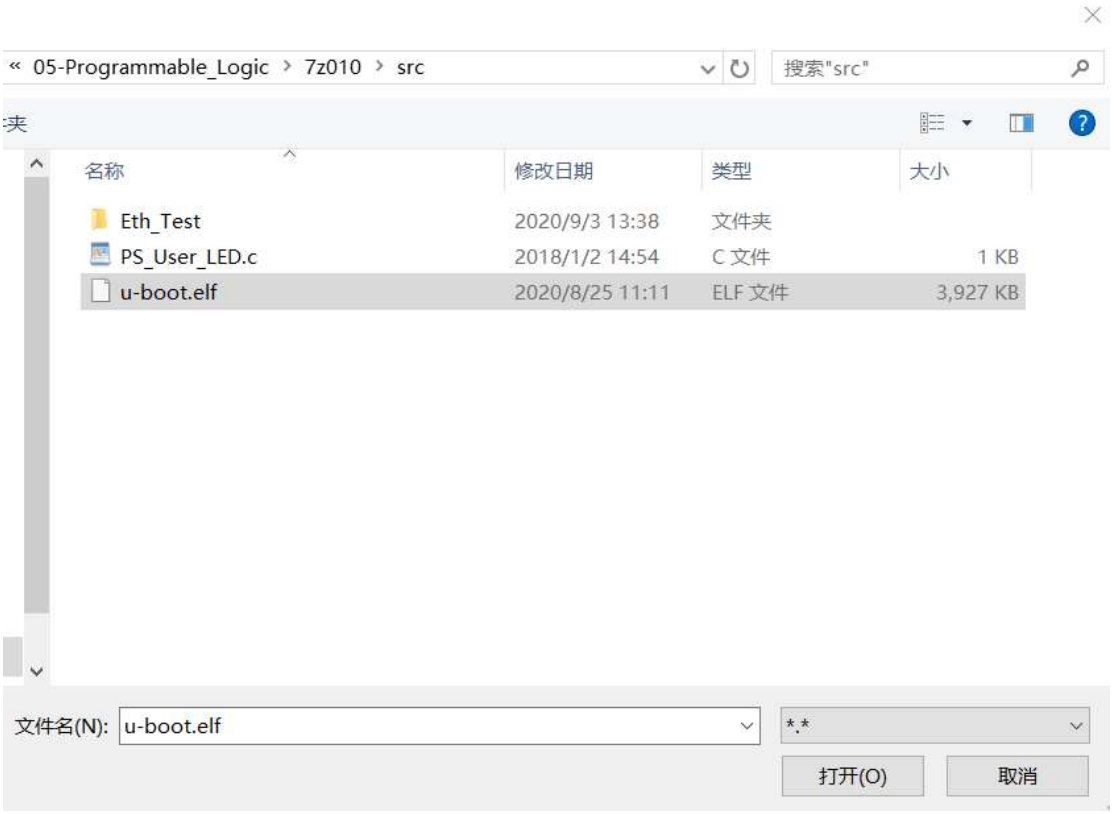
•As shown in figure below, click Add.



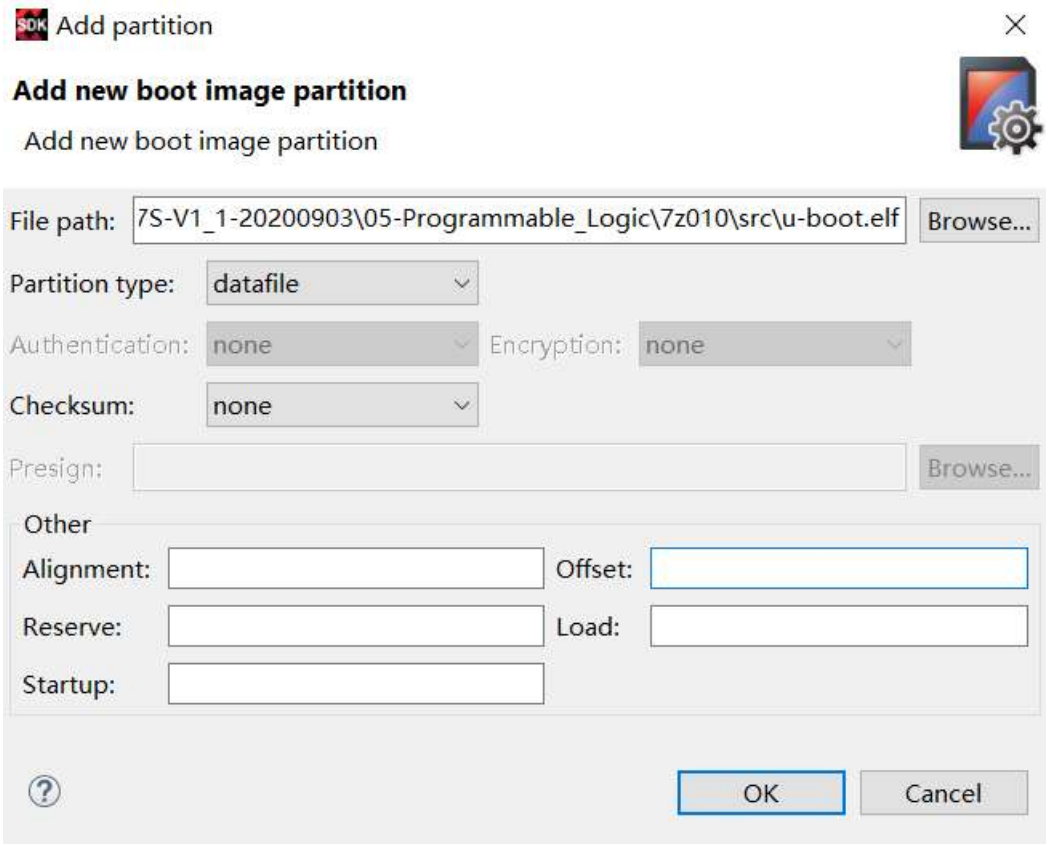
Click Browser



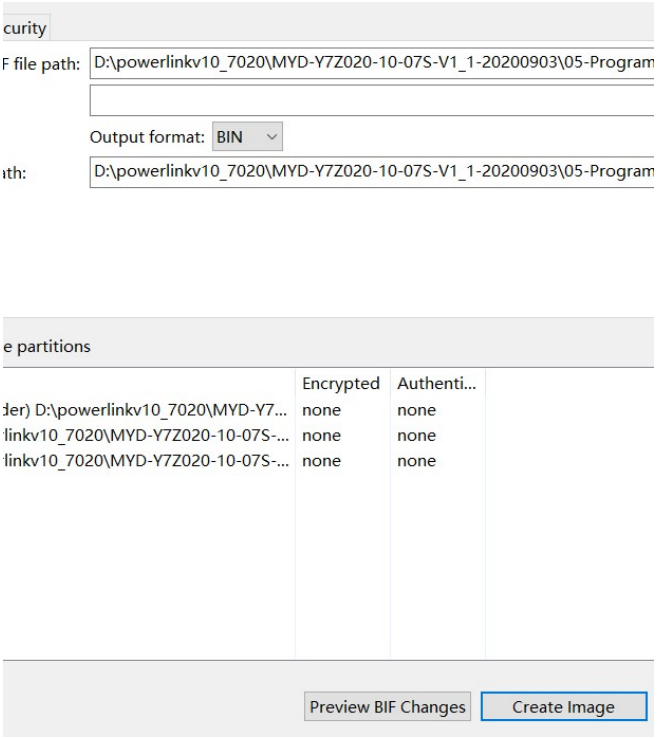
Choose u-boot.elf in the disk we have offered



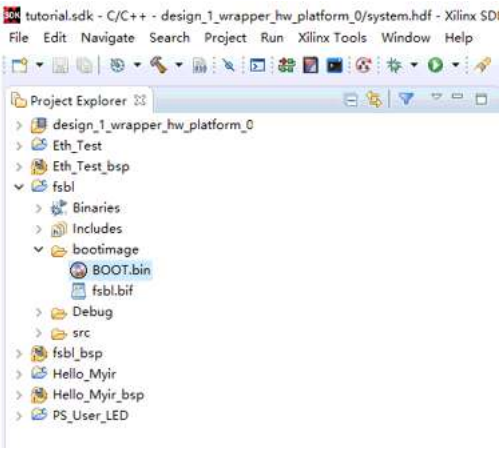
Click ok



Click Create Image



As shown in the figure below, you will find boot.bin in fsbl/bootimage. You can use this boot.bin with other files generated in linux development manual to run linux on MYD-Y7Z020/10/007S.



11 Setting up the Host PC

This section describes how to install the USB drivers on the host PC for the USB-UART connection to the MYD-Y7Z020/10/007S Starter Kit.

11.1Install the USB UART Drivers

Install the CDM21226_Setup.exe (Seriel Driver) in the disk (path:*\03-Tools\) on the host computer.

11.2Configure the Host Computer COM Port

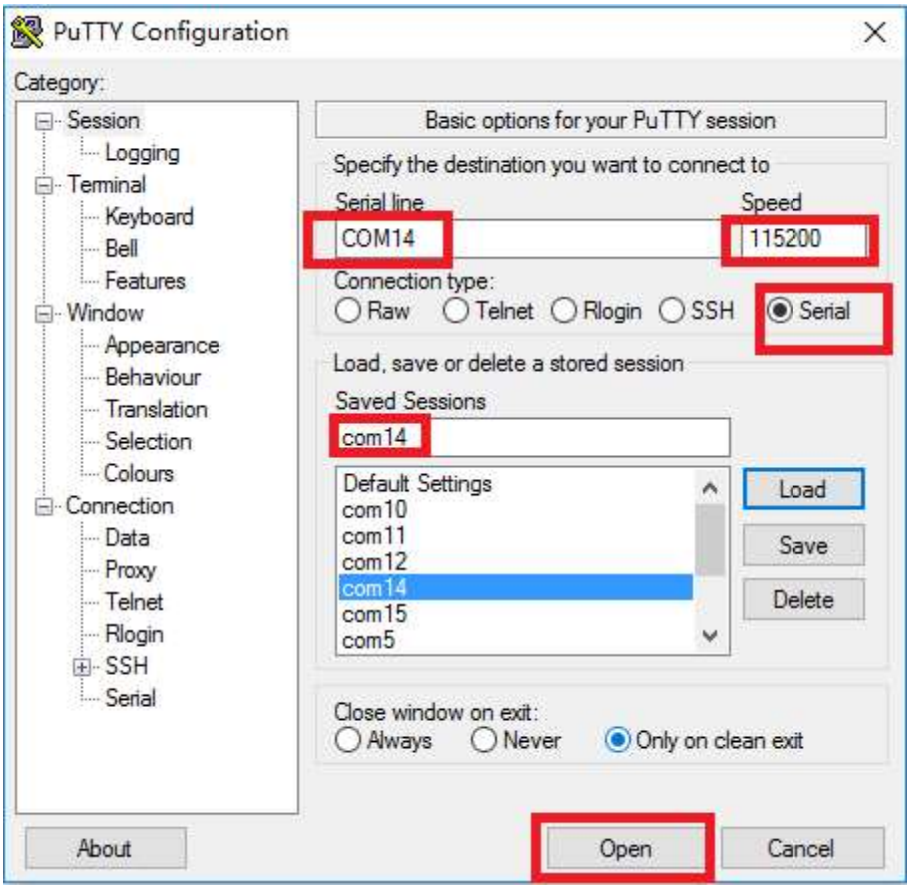
The Reference designs use a terminal program to communicate between the host computer and the MYD-Y7Z020/10/007S Starter Kit. To configure the host computer COM port for this purpose:

- Connect the MYD-Y7Z010/007S Starter Kit to the host computer via J8 Debug-UART port and power up the board.
- Open the host computer Device Manage as shown in the following figure. In the Windows task bar, click Start, click Control Panel, and then click Device Manager.
- Expand Ports (COM & LPT) , You will see USB Serial Port (COMXX). This is the COM which connect to MYD-Y7Z020/10/007S.


设备管理器
文件(F) 操作(A) 查看(V) 帮助(H)
DESKTOP-IVV8163
IDE ATA/ATAPI 控制器
Jungo
Programming cables
处理器
磁盘驱动器
存储控制器
打印队列
端口 (COM 和 LPT)
Prolific USB-to-Serial Comm Port (COM11)
USB Serial Port (COM14)
打印机端口 (LPT1)
通信端口 (COM1)
固件
计算机
监视器
软件设备
声音、视频和游戏控制器
通用串行总线控制器
网络适配器
系统设备
显示适配器
音频输入和输出

11.3Install the Terminal Program

Install the putty.exe terminal program in the disk (Path:*\03-Tools)on the host computer. To communicate with the MYD-Y7Z020/10/007S Starter Kit, configure the New Connection and Serial Port settings as shown in the following figure. These settings must match the host computer COM port settings shown in the previous section.

- 

Appendix I Disclaimer

This User Manual (the "Manual") is proprietary to MYIR Electronics Limited. ("MYIR") and no ownership rights are hereby transferred. No part of the Manual shall be used, reproduced, translated, converted, adapted, stored in a retrieval system, communicated or transmitted by any means, for any commercial purpose, including without limitation, sale, resale, license, rental or lease, without the prior express written consent of MYIR.

MYIR does not make any representations, warranties or guarantees, express or implied, as to the accuracy or completeness of the Manual. Users must be aware that updates and amendments will be made from time to time to the Manual. It is the user's responsibility to determine whether there have been any such updates or amendments. Neither MYIR nor any of its directors, officers, employees or agents shall be liable in contract, tort or in any other manner whatsoever to any person for any loss, damage, injury, liability, cost or expense of any nature, including without limitation incidental, special, direct or consequential damages arising out of or in connection with the use of the Manual.

Appendix II Technical Support and Warranty

MYIR Electronics Limited ("MYIR") is a global provider of ARM hardware and software tools, design solutions for embedded applications. We support our customers in a wide range of services to accelerate your time to market.

MYIR is an ARM Connected Community Member and work closely with ARM and many semiconductor vendors. We sell products ranging from board level products such as development boards, single board computers and CPU modules to help with your evaluation, prototype, and system integration or creating your own applications. Our products are used widely in industrial control, medical devices, consumer electronic, telecommunication systems, Human Machine Interface (HMI) and more other embedded applications. MYIR has an experienced team and provides custom design services based on ARM processors to help customers make your idea a reality.

The contents below introduce to customers the warranty and technical support services provided by MYIR as well as the matters needing attention in using MYIR's products.

Service Guarantee

MYIR regards the product quality as the life of an enterprise. We strictly check and control the core board design, the procurement of components, production control, product testing, packaging, shipping and other aspects and strive to provide products with best quality to customers. We believe that only quality products and excellent services can ensure the long-term cooperation and mutual benefit.

Price

MYIR insists on providing customers with the most valuable products. We do not pursue excess profits which we think only for short-time cooperation. Instead, we hope to establish long-term cooperation and win-win business with customers. So we will offer reasonable prices in the hope of making the business greater with the customers together hand in hand.

NCNR (Non-cancellation of orders and non-return of goods)

No returns or cancellations will be accepted without prior written agreement from MYIR.

Delivery Time

MYIR will always keep a certain stock for its regular products. If your order quantity is less than the amount of inventory, the delivery time would be within three days; if your order quantity is greater than the number of inventory, the delivery time would be always four to six weeks. If for any urgent delivery, we can negotiate with customer and try to supply the goods in advance.

Technical Support

MYIR has a professional technical support team. Customer can contact us by email

(support@myirtech.com), we will try to reply you within 48 hours. For mass production and customized products, we will specify person to follow the case and ensure the smooth production.

After-sale Service

MYIR offers 12 months free technical support and after-sales maintenance service from the purchase date. The service covers:

1. Technical support service

MYIR offers technical support for the hardware and software materials which have provided to customers;

To help customers compile and run the source code we offer;

To help customers solve problems occurred during operations if users follow the user manual documents;

To judge whether the failure exists;

To provide free software upgrading service.

However, the following situations are not included in the scope of our free technical support service:

Hardware or software problems occurred during customers' own development;

Problems occurred when customers compile or run the OS which is tailored by themselves;

Problems occurred during customers' own applications development;

Problems occurred during the modification of MYIR's software source code.

2. After-sales maintenance service

The products except LCD, which are not used properly, will take the twelve months free maintenance service since the purchase date. But following situations are not included in the scope of our free maintenance service:

The warranty period is expired;

The customer cannot provide proof-of-purchase or the product has no serial number;

The customer has not followed the instruction of the manual which has caused the damage the product;

Due to the natural disasters (unexpected matters), or natural attrition of the components, or unexpected matters leads the defects of appearance/function;

Due to the power supply, bump, leaking of the roof, pets, moist, impurities into the boards, all those reasons which have caused the damage of the products or defects of appearance;

Due to unauthorized weld or dismantle parts or repair the products which has caused the damage of the products or defects of appearance;

Due to unauthorized installation of the software, system or incorrect configuration or computer virus which has caused the damage of products.

Warm tips:

MYIR does not supply maintenance service to LCD. We suggest the customer first check the LCD when receiving the goods. In case the LCD cannot run or no display, customer should contact MYIR within 7 business days from the moment get the goods.

Please do not use finger nails or hard sharp object to touch the surface of the LCD.

MYIR suggests user purchasing a piece of special wiper to wipe the LCD after long time use, please avoid clean the surface with fingers or hands to leave fingerprint.

Do not clean the surface of the screen with chemicals.

Please read through the product user manual before you using MYIR's products.

For any maintenance service, customers should communicate with MYIR to confirm the issue first. MYIR's support team will judge the failure to see if the goods need to be returned for repair service, we will issue you RMA number for return maintenance service after confirmation.

3. Maintenance period and charges

a) MYIR will test the products within three days after receipt of the returned goods and inform customer the testing result. Then we will arrange shipment within one week for the repaired goods to the customer. For any special failure, we will negotiate with customers to confirm the maintenance period.

b) For products within warranty period and caused by quality problem, MYIR offers free maintenance service; for products within warranty period but out of free maintenance service scope, MYIR provides maintenance service but shall charge some basic material cost; for products out of warranty period, MYIR provides maintenance service but shall charge some basic material cost and handling fee.

4. Shipping cost

During the warranty period, the shipping cost which delivered to MYIR should be responsible by user; MYIR will pay for the return shipping cost to users when the product is repaired. If the warranty period is expired, all the shipping cost will be responsible by users.

5. Products Life Cycle

MYIR will always select mainstream chips for our design, thus to ensure at least ten years continuous supply; if meeting some main chip stopping production, we will inform customers in time and assist customers with products updating and upgrading.

Value-added Services

MYIR provides services of driver development base on MYIR's products, like serial port, USB, Ethernet, LCD, etc.

MYIR provides the services of OS porting, BSP drivers' development, API software development, etc.

MYIR provides other products supporting services like power adapter, LCD panel, etc.

ODM/OEM services.

Appendix III Contact Us

MYIR Electronics Limited

Address:
Room 05, 6th Floor, Building No.2, Fada Road,
Yunli Intelligent Park, Bantian, Longgang District,
Shenzhen, Guangdong,
China 518129

Phone: +86-755-22984836
Fax: +86-755-25532724
Sales E-mail:sales@myirtech.comor myirtech@yahoo.com
Support E-mail:support@myirtech.com(support)
Website:http://www.myirtech.com

Thanks for using MYIR's products and services. For more information, please refer to the website or contact MYIR.