MicroZed: Creating a Zynq Hardware Platform in Vivado

02 June 2014
Version 2014_1.01

Overview

With a traditional processor, the hardware platform is pre-defined. The manufacturer selected the processor parameters and built-in peripherals when the chip was designed. To make use of this pre-defined processor, you need only target that specific hardware platform in the software development tools.

The Zynq-7000 All Programmable SoC is different. Zynq provides multiple building blocks and leaves the definition to you as the design engineer. This adds flexibility, but it also means that a bit of work needs to be done up front before any software development can take place.

The first step in completing a Zynq design is to define and build the hardware platform. The purpose of this tutorial is to show you how to quickly and easily create a base hardware platform for MicroZed.

Objectives

When this tutorial is complete, you will be able to:

- Create a new project in Vivado, targeted at MicroZed 7010
- Create a block based design to insert an ARM processor core
- Import the MicroZed Zynq PS Preset settings
- Build and export the hardware platform

© 2014 Avnet. All rights reserved. All trademarks and registered trademarks are the property of their respective owners. All specifications are subject to change without notice.

NOTICE OF DISCLAIMER: Avnet is providing this design, code, or information "as is." By providing the design, code, or information as one possible implementation of this feature, application, or standard, Avnet makes no representation that this implementation is free from any claims of infringement. You are responsible for obtaining any rights you may require for your implementation. Avnet expressly disclaims any warranty whatsoever with respect to the adequacy of the implementation, including but not limited to any warranties or representations that this implementation is free from claims of infringement and any implied warranties of merchantability or fitness for a particular purpose.

Experiment Setup

Software

The software used to test this reference design is:

- Windows-7 64-bit
- Xilinx Vivado 2014.1
- MicroZed Board Definition Install for Vivado 2014.1
    - http://www.microzed.org/documentation/1519

Hardware

The hardware setup used to test this reference design includes:

- Win-7 PC with a recommended 1.6 GB RAM available for the Xilinx tools to complete a XC7Z010 design[1]

[1] Refer to www.xilinx.com/design-tools/vivado/memory.htm

Experiment 1: Create a New Zynq Project in Vivado

The MicroZed Evaluation Kit includes a license voucher for Vivado Design Edition, device-locked to the Z7010 device. Vivado WebPack also supports MicroZed. The Zynq Processing System (PS) may be used without anything programmed in the Programmable Logic (PL). This PS-only style is the simplest way to use Zynq, so that is what we will do during this lab. However, the power of Zynq is found in using soft IP in the PL, interconnecting PS to PL, and routing extra PS built-in peripherals through EMIO to PL I/Os, and then programming of the PL is required.

This tutorial will take advantage of built-in 3rd-party board definition files. The MicroZed Board definition archive is posted at www.microzed.org → Documentation → MicroZed → *MicroZed Board Definition Install for Vivado 2014.1*

1. If not previously completed, download the MicroZed Board Definition Install for Vivado 2014.1 archive and follow the instructions to install the board definitions.

2. Launch Vivado by selecting **Start → All Programs → Xilinx Design Tools → Vivado 2014.1 → Vivado 2014.1**.

3. If you are using a User Repository, use the TCL Console to set the Repository Path, similar to the following:

```
set_param project.boardPartRepoPaths C:/AvnetBoards
```

4. Select **File → New Project** or click on **Create New Project** under *Quick Start*.

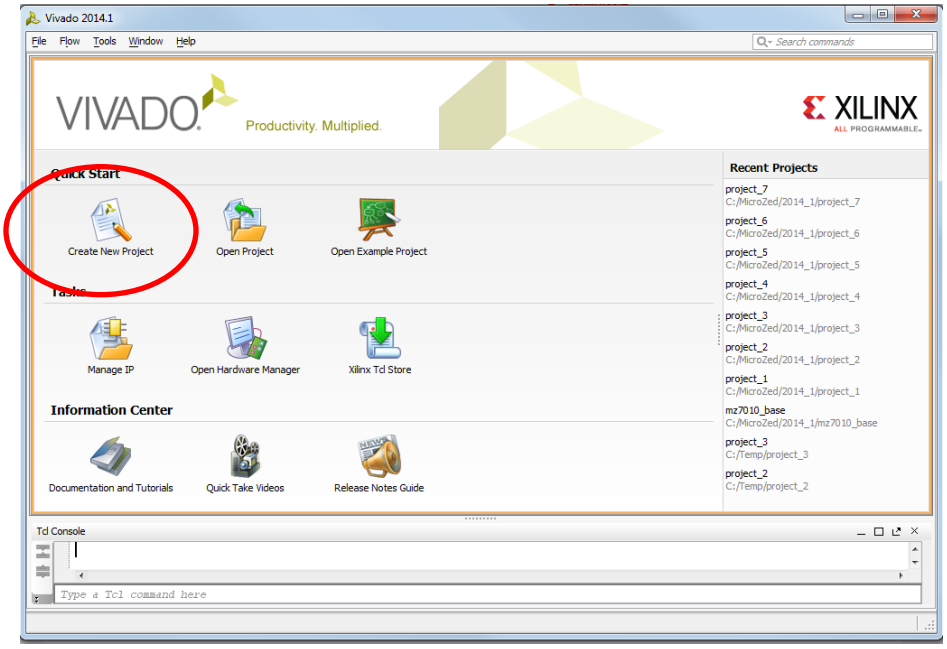


Figure 1 – Vivado Launched

5. Click **Next >**.

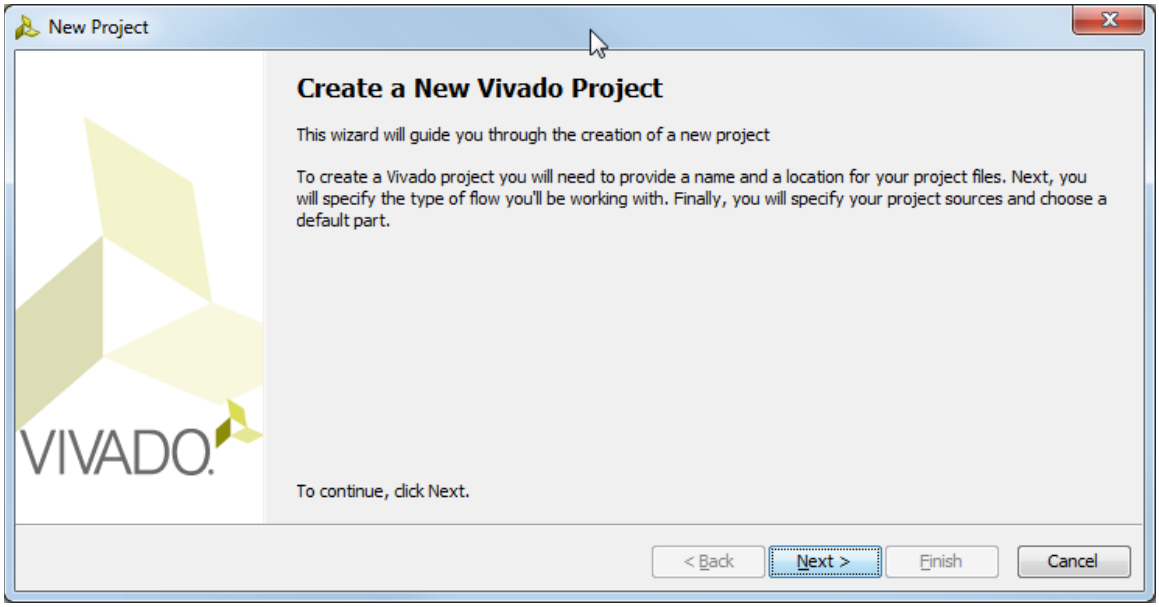


Figure 2 – New Vivado Project Wizard launched

6. Click the browse icon . Browse to set the *Project location* to your desired project location and click **Select**.

7. Set the *Project name* to **`MZ_Basic_System`**. Also verify the ***`Create project subdirectory`*** checkbox is selected. Click **Next >**.

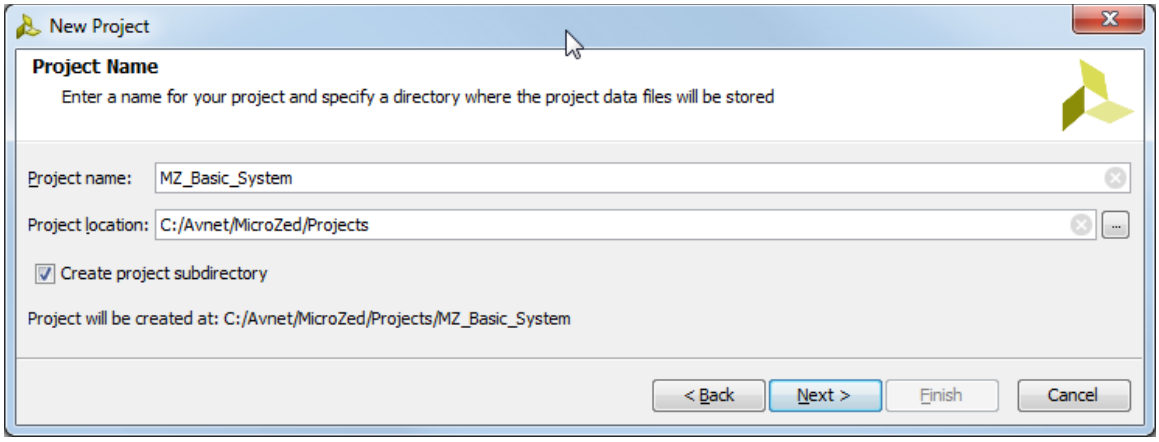


Figure 3 – Set Project Name and Location

8. The project will be RTL based, so leave the radio button for *RTL Project* selected. Since this is a brand new project, check the box for ***Do not specify sources at this time***. Click **Next >**.Click **Next >**.

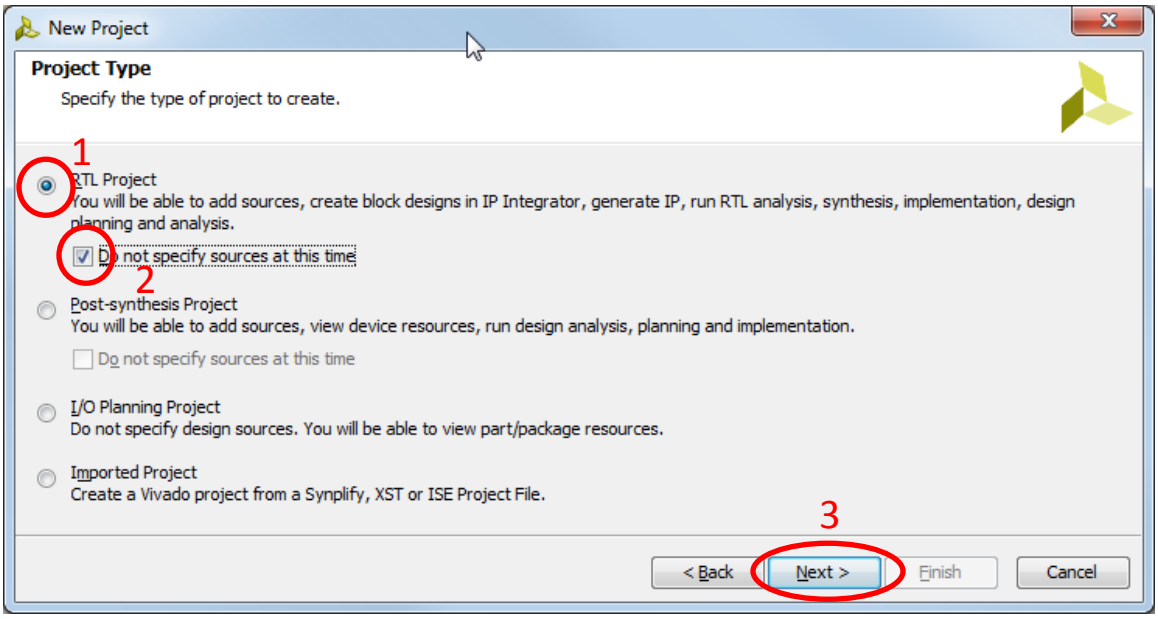


Figure 4 – Set Project Type

Next, the Default Part is selected. This can be done by specifying a specific part or by selecting a board. A generic "microzed" for the 7010 would normally show up in the Boards list, but not the 7020. If you have installed the MicroZed Board Definition archive correctly, both the 7010 and 7020 MicroZed boards will show up.

9. In the *Specify* area, select **Boards**.

10. Set the *Board Vendor* to **em.avnet.com**. This should leave only three boards in the table.

11. Single-click the **MicroZed 7010 Board**. Note that the Rev F board may be selected for any revision B, C, or F. Click **Next >**.

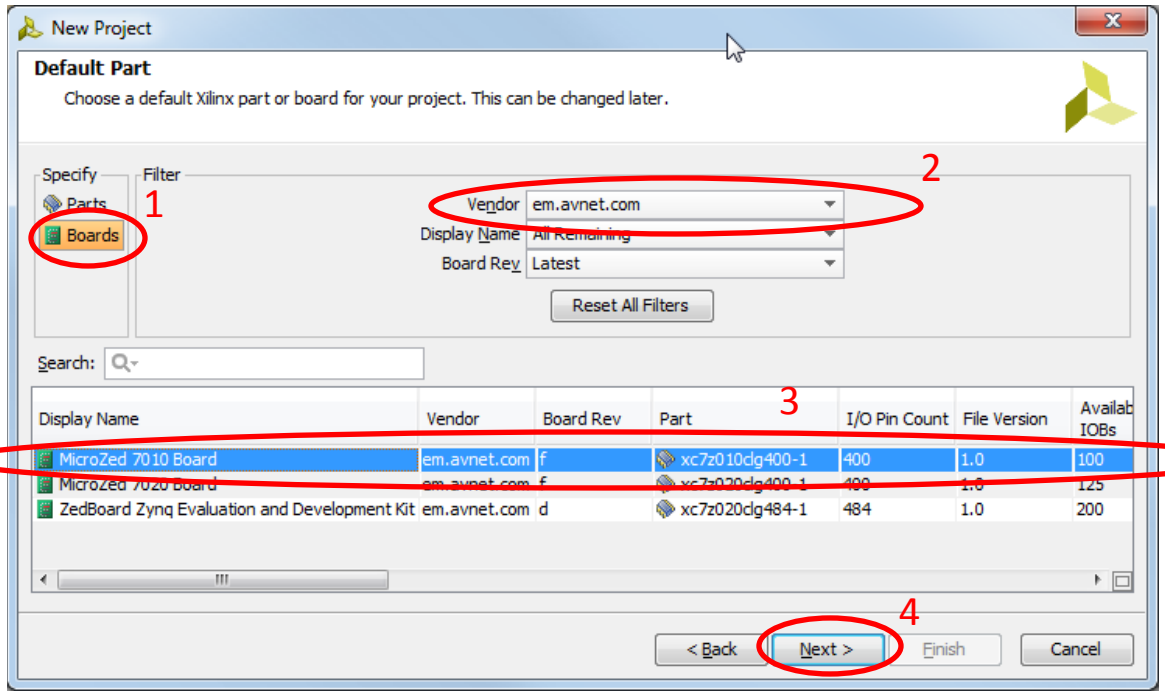


Figure 5 – Select the Target Board = MicroZed 7010

12. A project summary is displayed. Click **Finish**. The Vivado cockpit is now displayed.

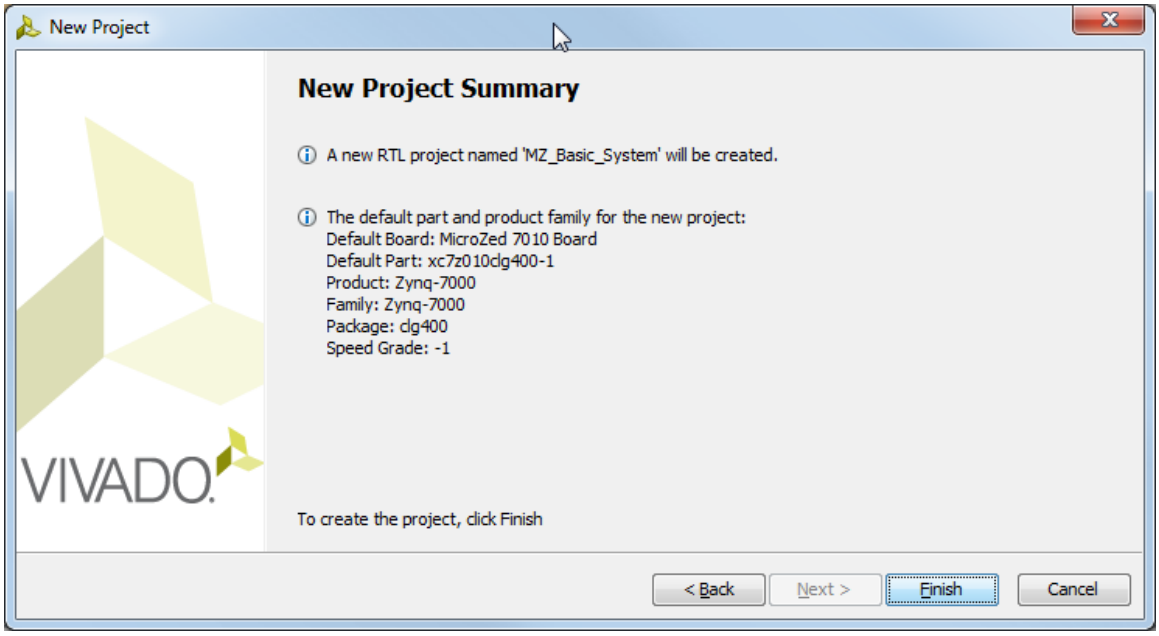


Figure 6 – New Project Summary

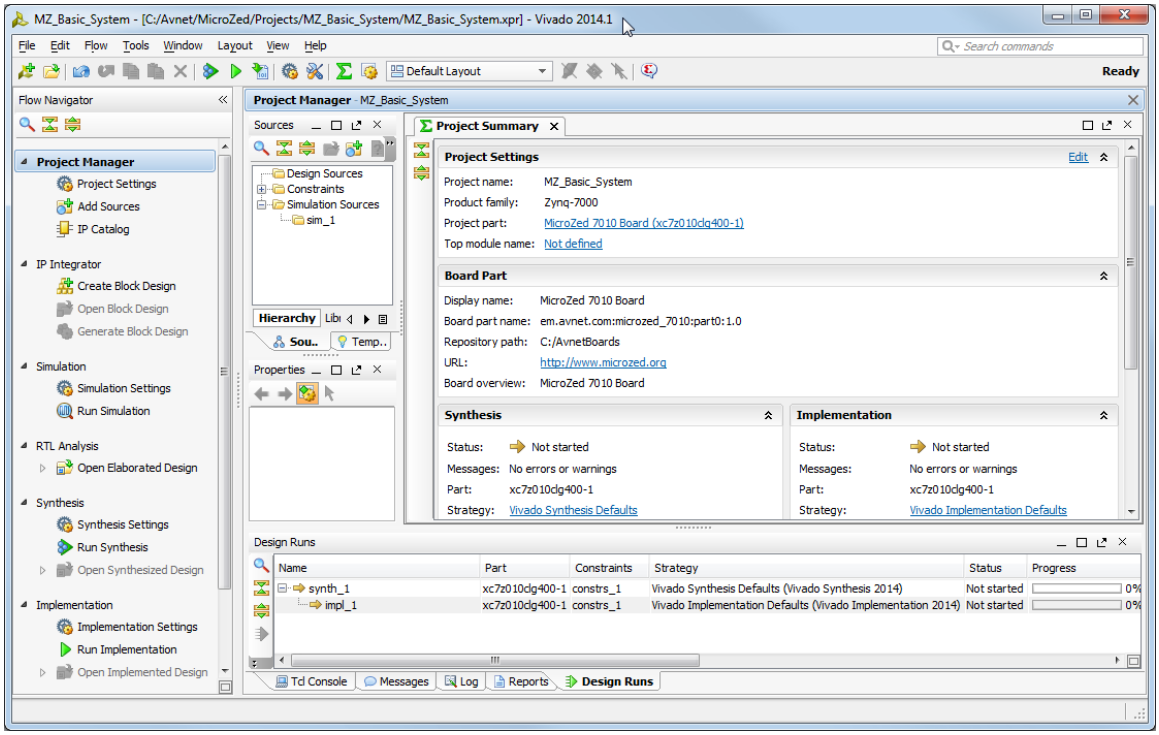


Figure 7 – Vivado Cockpit

Experiment 2: Create and Edit a Block Design

The current project is blank. To access the ARM processing system, we will add an embedded source to the Vivado project using IP Integrator.

1. The recommended way to add an embedded processor is through the Block Design method via IP Integrator. Select **Create Block Design**.

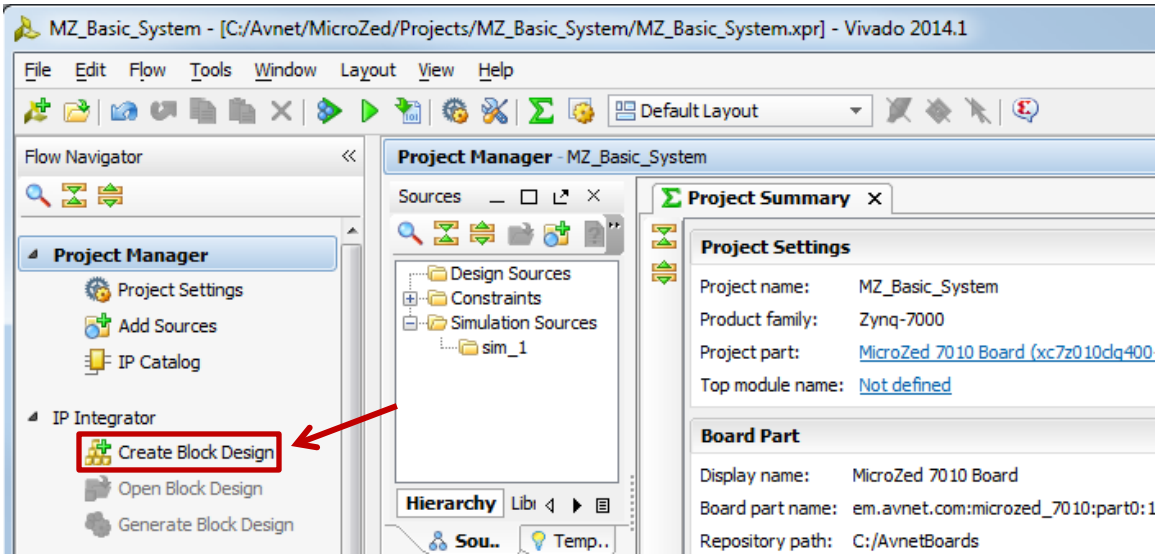


Figure 8 - Create Block Design

2. Give the Block Design a name. *System* is commonly used. Click **OK**.

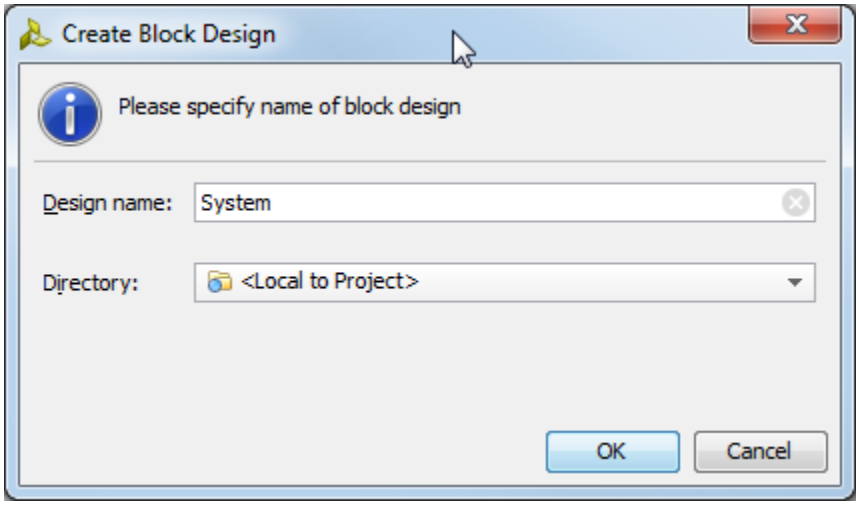


Figure 9 - Block Design Name

3. In the Diagram window, click either the **Add IP** text or icon .

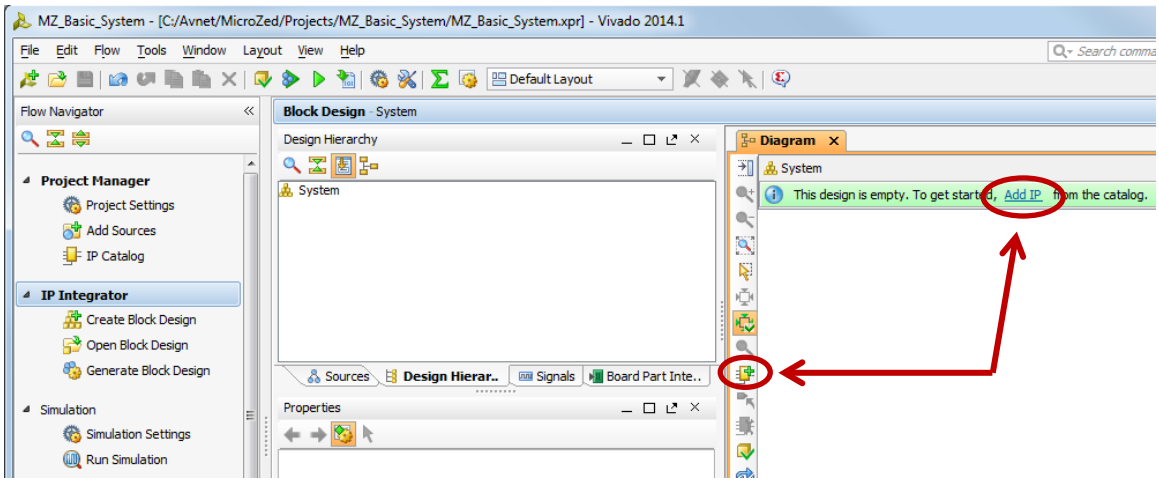


Figure 10 – Add IP to the Block Design

4. The *Add Sources* window opens. Start typing "Zynq" in the search window. Find the **ZYNQ7 Processing System** IP. Either double-click this or drag and drop to the *Diagram* window.

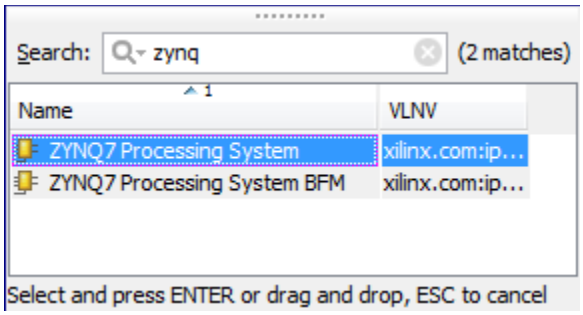


Figure 1 – Add IP Window

The Zynq Processing system will appear in the *Diagram* window. Also a new tab will appear labeled **Address Editor**.

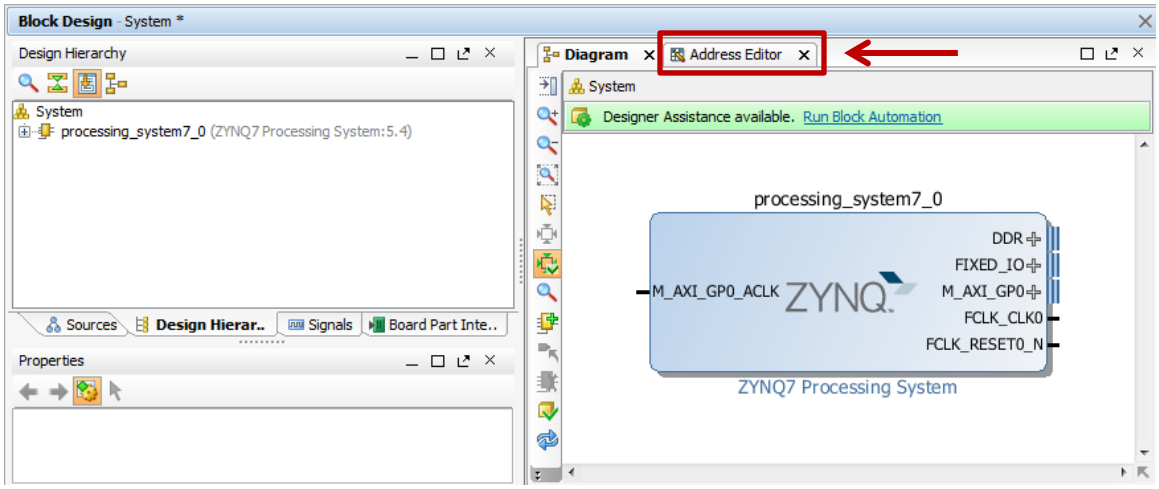


Figure 2 – Updated Block Diagram

5. Similar to the *Add IP* prompt in the previous step, notice now that the *Designer Assistance* has provided the hint to *Run Block Automation*. Click the **Run Block Automation** link at the top of the window and select **/processing_system7_0** to connect all block I/O.

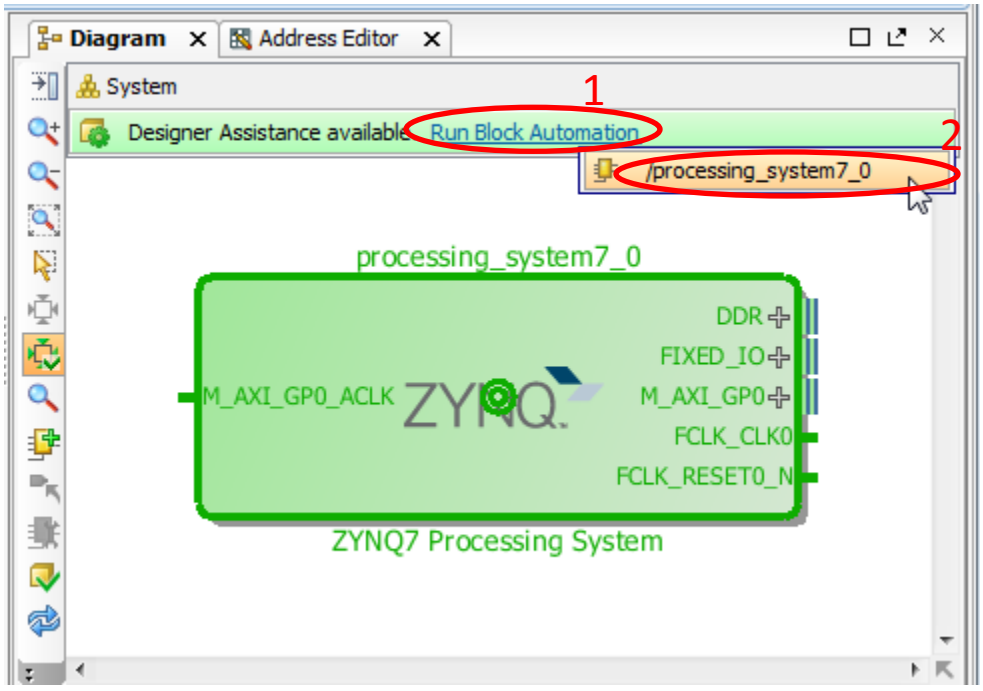


Figure 3 - Run Block Automation

6. Notice the block automation wizard has identified two sources of I/O that need to be made external. One is obvious, the **DDR** interface. The other is labeled **FIXED_IO**. FIXED_IO is basically the MIO pin connections. They are labeled FIXED_IO because you cannot change their assignments in this window.

   The *Apply Board Preset* checkbox applies the Preset TCL that was included as part of the board definition archive. Leave this checked. For details about how to build a system manually, please see the *Avnet 2013.3 Zynq Hardware Development Speedway*.

   The Cross Trigger options may be left Disabled.

   Click **OK** to connect these external signals.

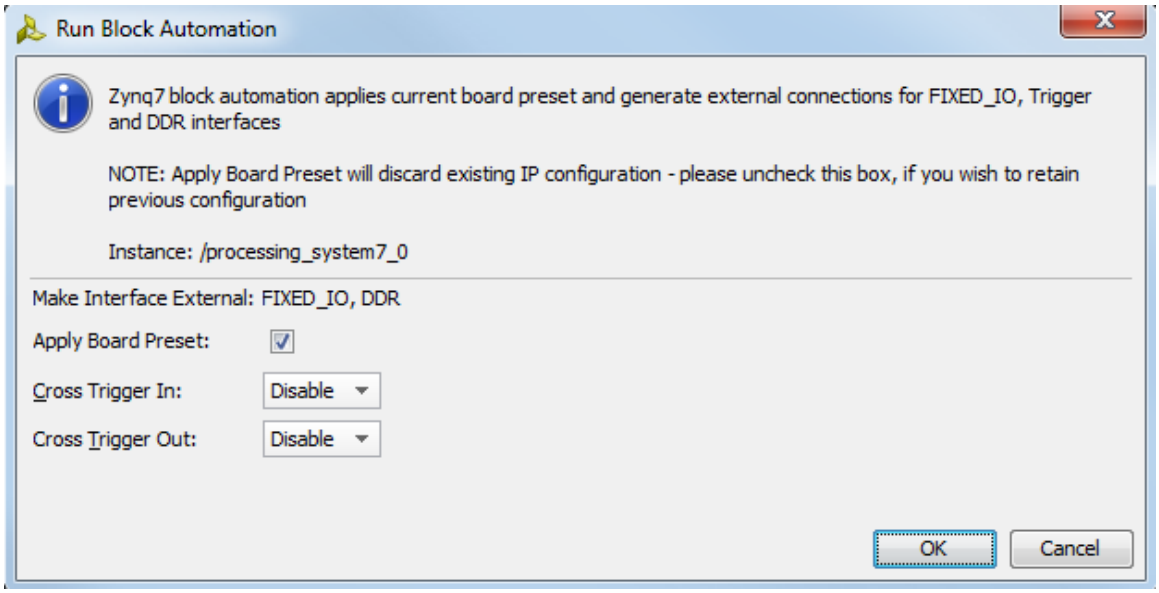


Figure 4 – Run Block Automation

7. You will now see the Zynq block with external I/O. This is much easier to visualize than the old XPS Netlist view.

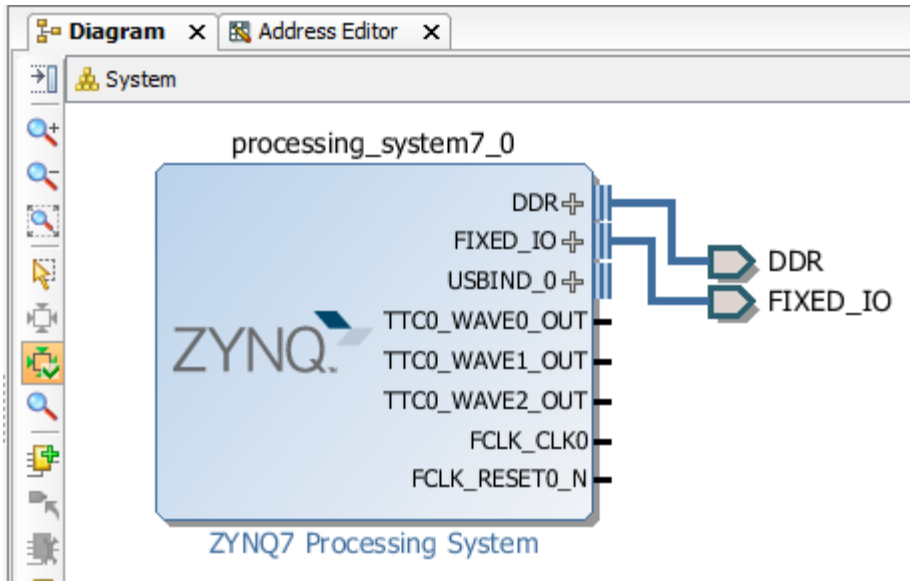


Figure 5 - Zynq Block Diagram with External I/O

8. At this point, we can **validate** our design. Click the Validate Design icon . A successful validation window will appear. Click **OK**.

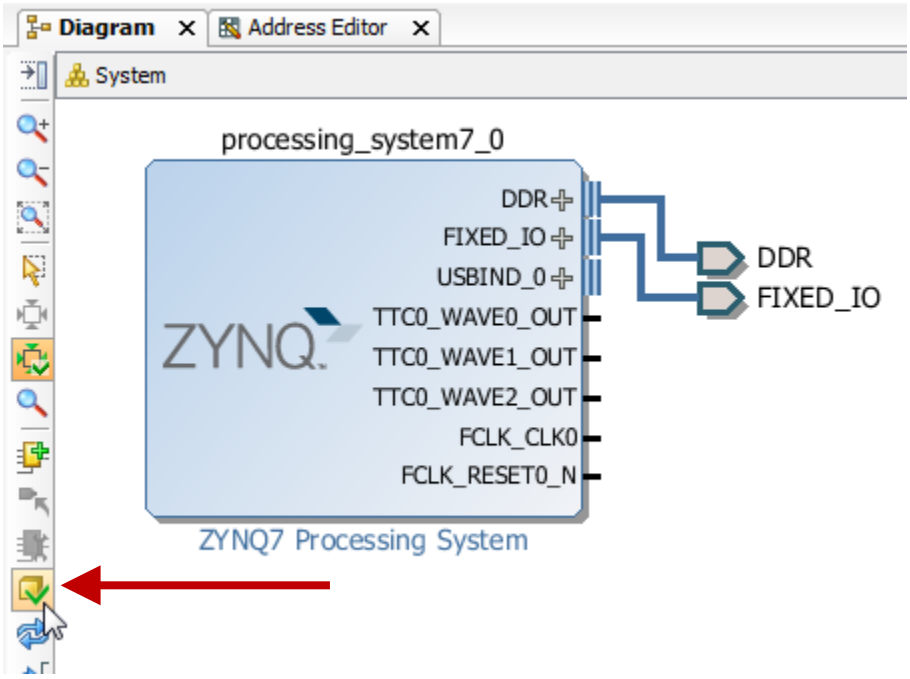


Figure 6 - Validate Zynq Block Design

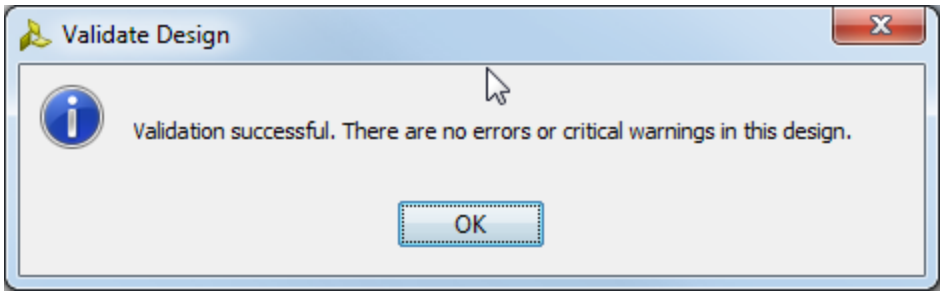
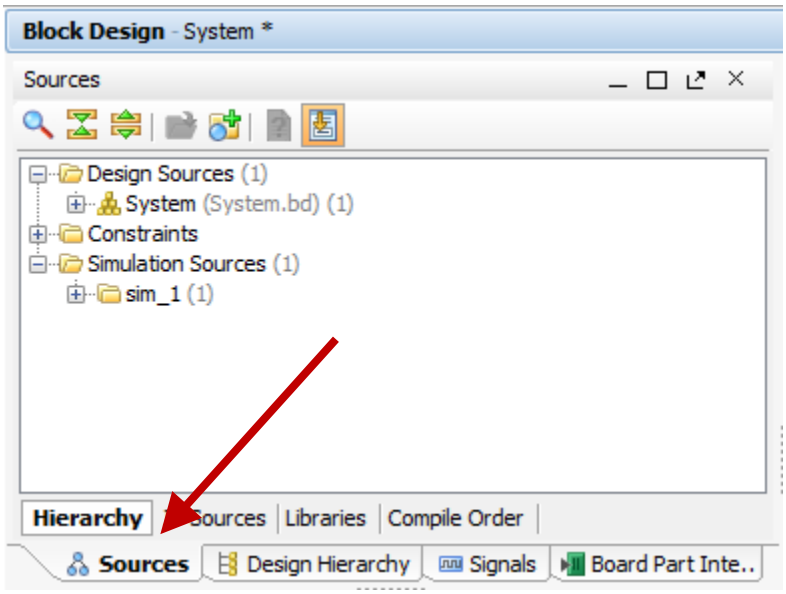


Figure 7 – Validation Successful

9. The next three steps are not intuitive, but must be done. We must generate output products of our Zynq block design. To do this, we need to switch back to the **Sources** tab.



10. Click **Save Block Design** icon, , to save the project.

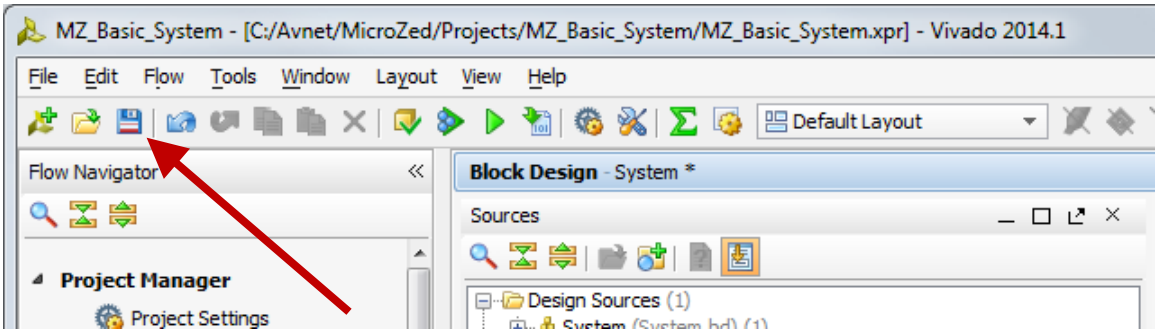


Figure 8 - Sources Tab

11. In the Sources Window, right-click on **System.bd** and select **Generate Output Products…**

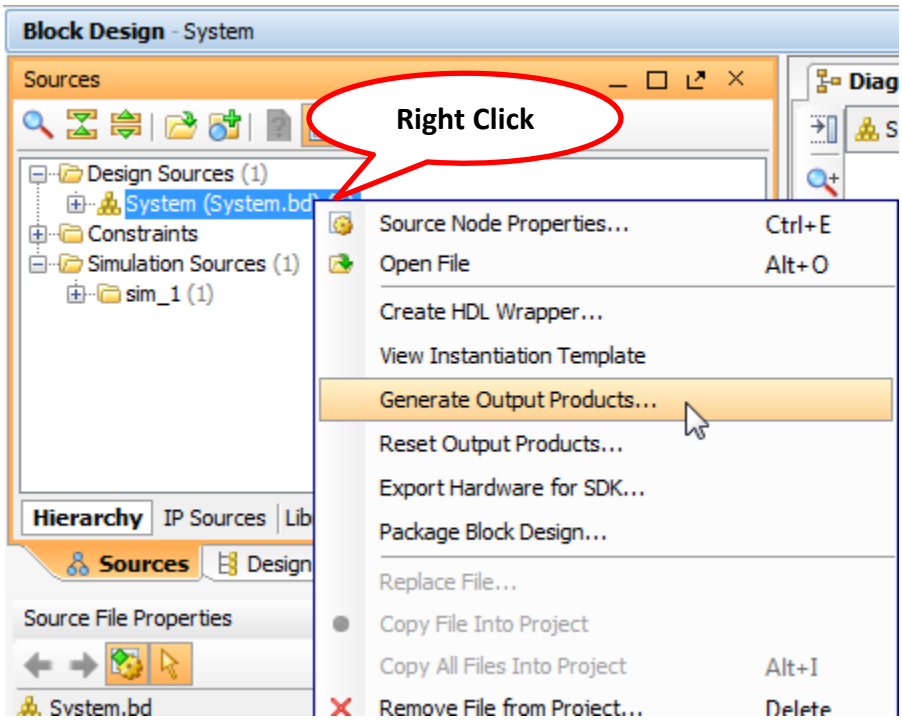


Figure 9 - Generate Output Products

In the *Generate Output Products* window, you can see what output products will be generated. It does not actually run through synthesis or implementation; it just creates the necessary files for Synthesis and Implementation, similar to the *Create Netlist* function from the old ISE-based XPS flow.

12. Click **Generate**

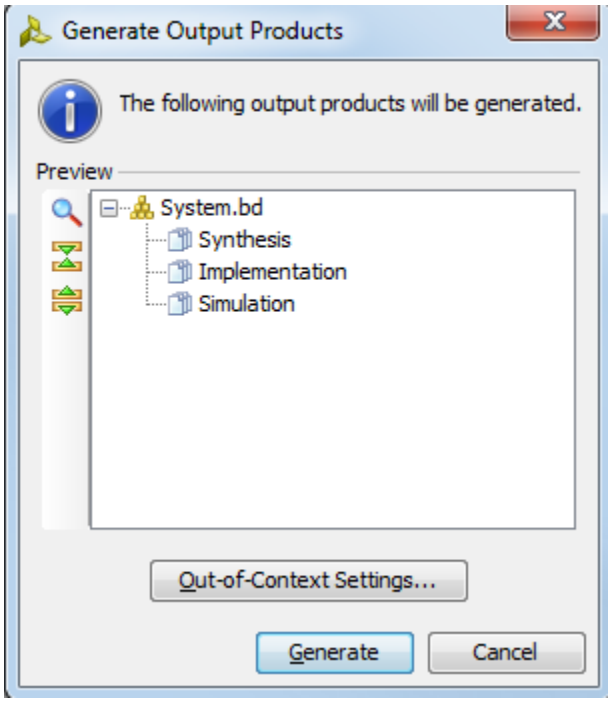


Figure 10 – Manage Output Products

13. Once completed right-click on **System.bd** again and select **Create HDL wrapper**.

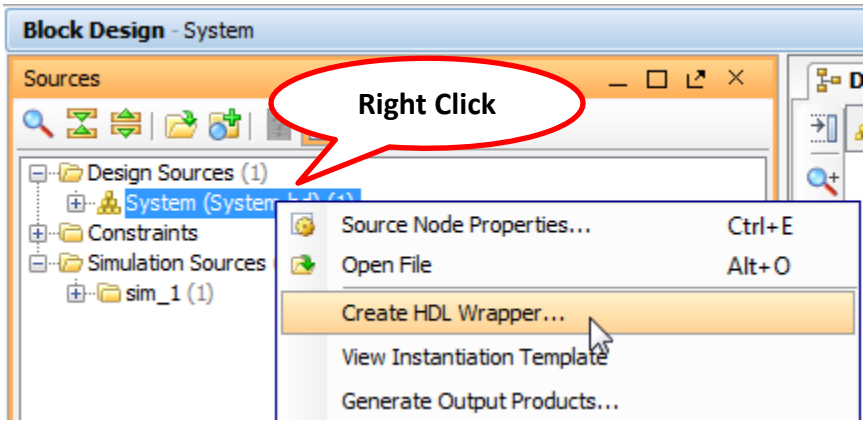


Figure 11 - Create Top Level HDL Wrapper

14. For now, leave the option selected to *Let Vivado manage wrapper and auto-update*. Click **OK**.

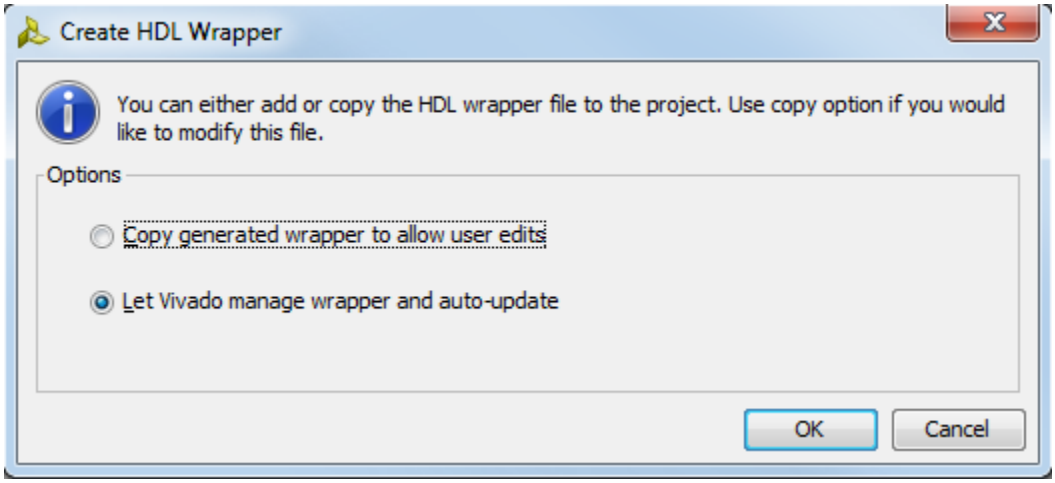


Figure 12 – Let Vivado Manage Wrapper

15. Once the top-level wrapper is created, you can see the design hiearchy in the *Sources* tab. Notice that **System_wrapper.v** is the top-level HDL wrapper that was created. **System.bd** is the Block Design.

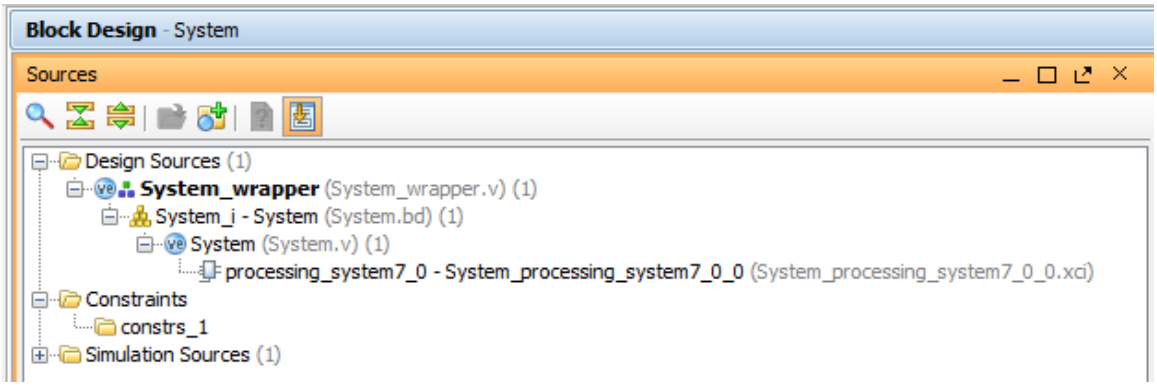


Figure 13 – System_wrapper.v Generated

16. Click **Generate Bitstream** in the *Flow Navigator* window. Click **Yes** to start Synthesis and Implementation flows. *Check the upper right-hand corner of the tool for a status bar.*

17. When bitstream generation is completed, click **OK** to *Open Implemented Design*.

Experiment 3: Export Hardware Platform to SDK

Now that we've created an embedded system, we must make this platform available to the Software Development Kit (SDK). This is done by exporting the hardware platform.

1. In the Vivado tool, select **File → Export → Export Hardware for SDK...**

   You could check the box for **Launch SDK** if you were also ready to start working on the software application design right now. However, as it may be common for a hardware engineer to create the hardware platform and hand it over to a software engineer, this tutorial will simply export the hardware platform as if it were being handed over to a colleague or stored for later use.

   Make sure the **Export Hardware** box is checked. Click **OK.**

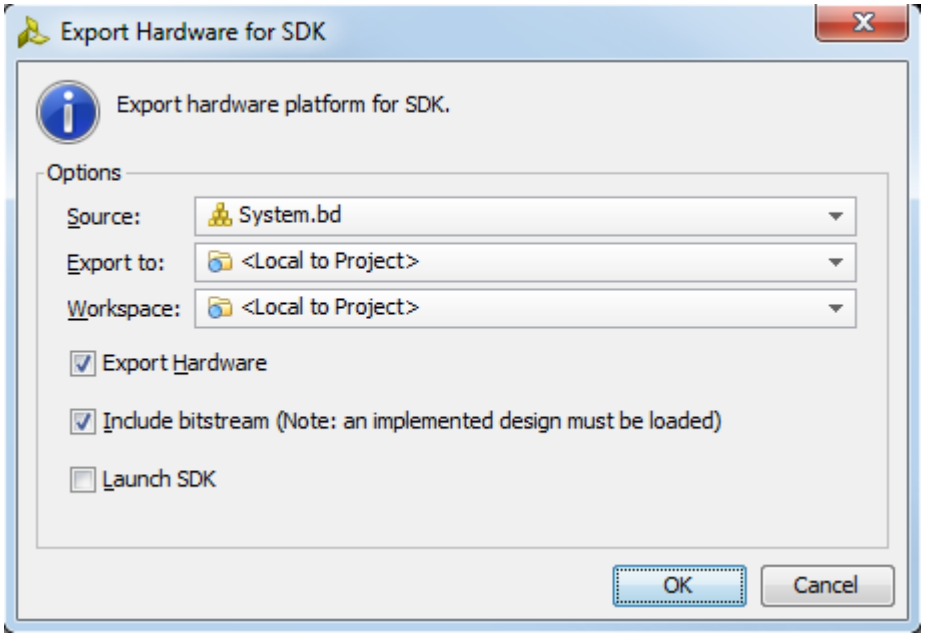


Figure 14 – Export Zynq hardware platform to SDK

2. We will now explore what you have created. In Windows Explorer, browse to your project directory.

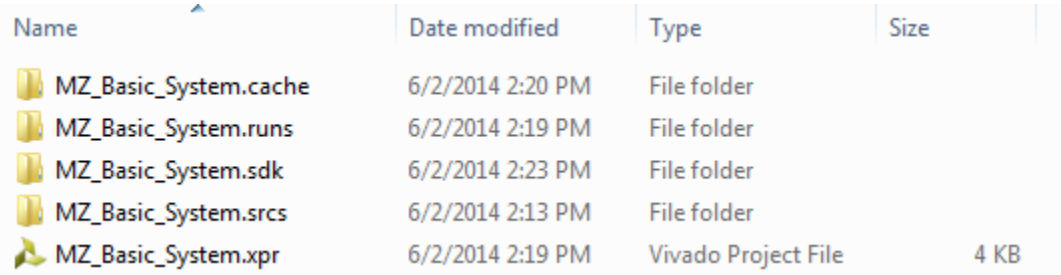


Figure 15 – Project Directory Contents after Export to SDK

You will notice four directories and one file here. The .xpr file is your Vivado Project File and can be used to re-launch your project when you come back to work on it some more.

The .cache, .runs, and .srcs directories contain everything related to the hardware design, including the block design source, wrapper HDL, and synthesis/implementation results.

The .sdk folder is the result of the ***Export Hardware for SDK*** operation. Everything required for SDK to import the hardware platform is contained inside this directory. A hardware engineer looking to share the design with the software team could provide this one folder. This provides a very compact and portable method to send a Zynq Hardware Platform to a colleague.

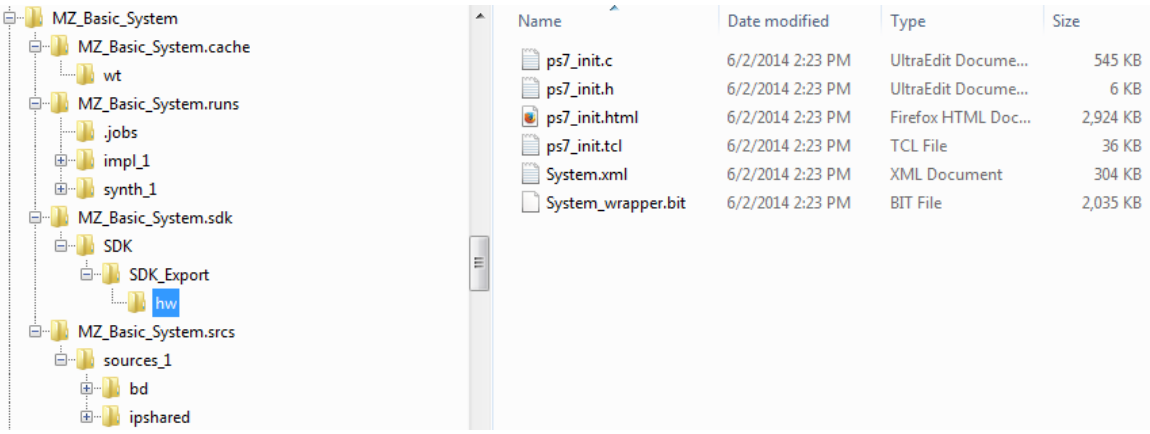


Figure 16 – Zynq Hardware Platform Export for SDK

Revision History

Date	Version	Revision
23 Aug 2013	2013_2.01	Initial Avnet release for Vivado 2013.2
02 Jun 2014	2014_1.01	Update for 2014.1 using Avnet MicroZed board definition archive.