

## 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:

- Install the MicroZed board awareness file
- 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

---

## Experiment Setup

### Software

The software used to test this reference design is:

- Windows-7 64-bit
- Xilinx Vivado 2013.2
- *MicroZed Board Awareness archive for Vivado 2013.2*
  - [www.microzed.org](http://www.microzed.org) → Documentation → MicroZed
- *MicroZed Zynq PS Preset for Vivado 2013.2 (TCL)*
  - [www.microzed.org](http://www.microzed.org) → Documentation → MicroZed

### 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<sup>1</sup>

---

<sup>1</sup> Refer to [www.xilinx.com/design-tools/vivado/memory.htm](http://www.xilinx.com/design-tools/vivado/memory.htm)

## Experiment 1: Install the MicroZed board awareness file

Vivado 2013.2 has awareness of several Xilinx development boards built-in, as well as the Avnet ZedBoard. 2013.2 does not have full support for 3<sup>rd</sup>-party boards built-in, but it does have partial support. To take advantage of this, an XML file must be copied to the proper location for Vivado to find it.

1. Download the MicroZed Board Awareness archive.
2. Unzip to `C:/Xilinx/Vivado/2013.2/data/boards/zynq`. When complete, this directory should appear as shown below.

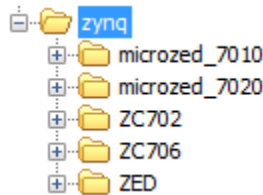


Figure 1 – MicroZed Board Awareness Archive Installed

Be aware that once you create a project based on these added boards, the Board Awareness files become a dependency on that project. Therefore, if you share that project with a colleague, they will also need to install the same Board Awareness archive or they will get an error regenerating your project.

## Experiment 2: 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.

1. Launch Vivado by selecting **Start → All Programs → Xilinx Design Tools → Vivado 2013.2 → Vivado 2013.2**.
2. Click on **Create New Project**.



Figure 2 – Vivado Launched

3. Click **Next >**.

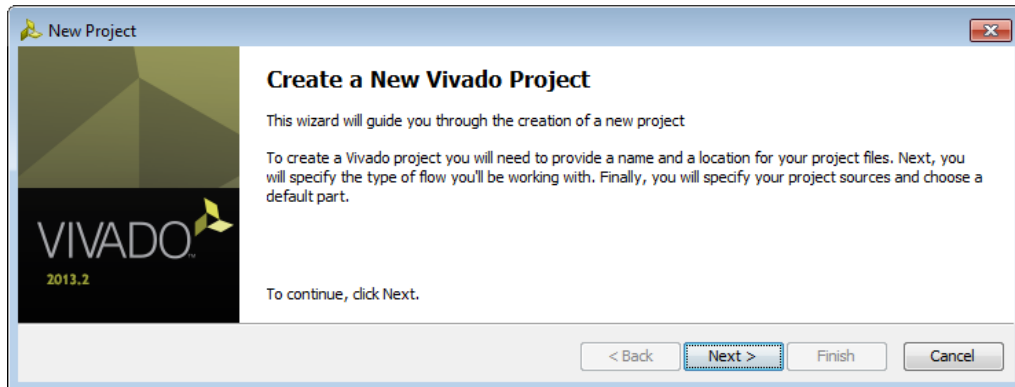



Figure 3 – New Vivado Project Wizard launched

4. Click the browse icon . Browse to set the *Project location* to your desired project location and click **Select**.
5. Set the *Project name* to **MZ\_Basic\_System**. Also verify the **Create project subdirectory** checkbox is selected. Click **Next >**.

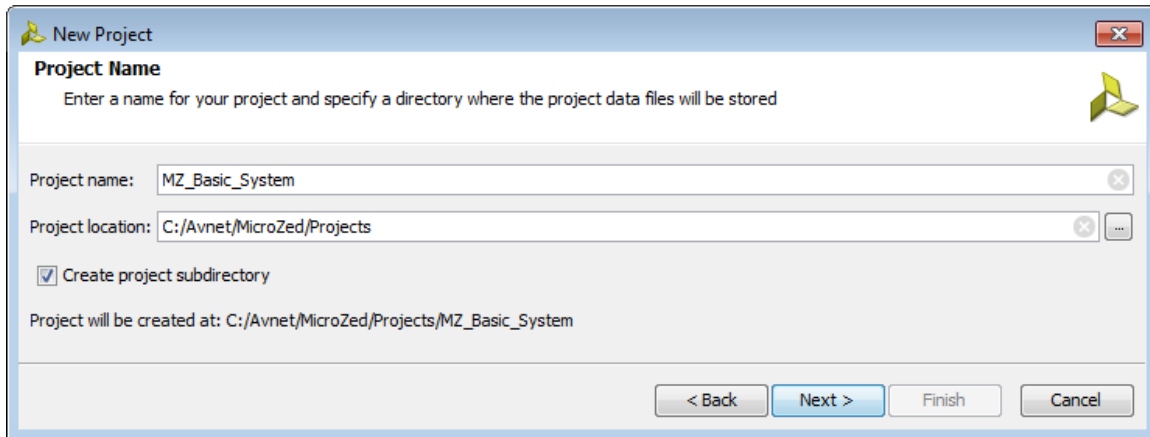
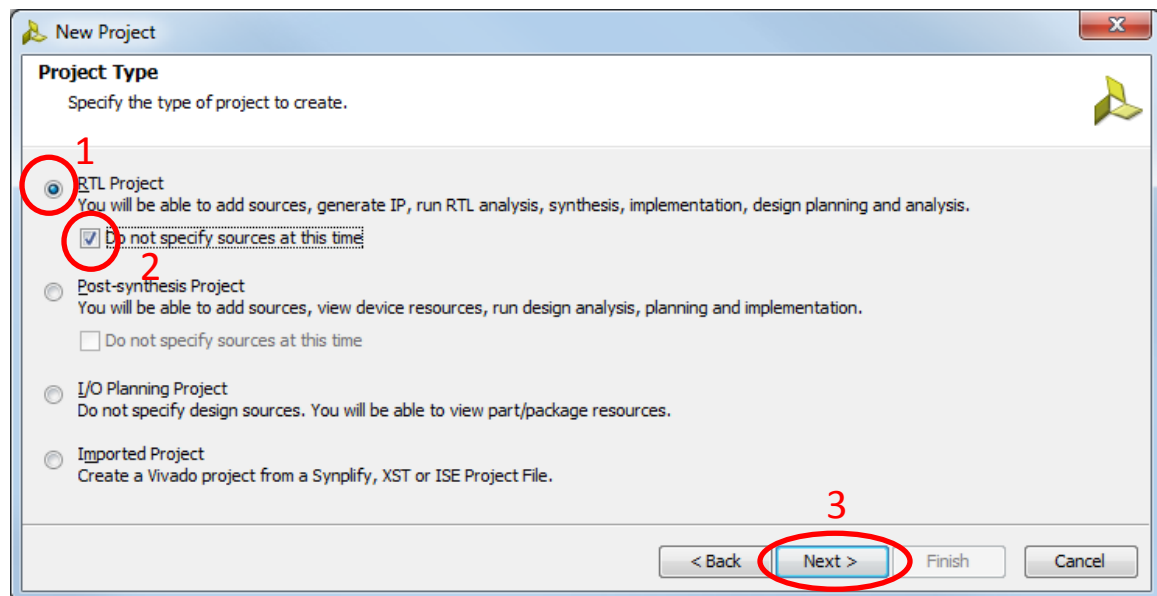


Figure 4 – Set Project Name and Location

6. 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 >**.



7. Figure 5 – Set Project Type

Next, the Default Part is selected. This can be done by specifying a specific part or by selecting a board. MicroZed would normally not show up in the Boards list. However, if you have installed the MicroZed Board Awareness archive correctly, it will.

8. In the *Specify* area, select **Boards**.
9. Set the *Board Vendor* to **em.avnet.com**. This should leave only six boards in the table.
10. Single-click the **MicroZed 7010 Board**, Rev B or C, depending on which board you have. Click **Next >**.

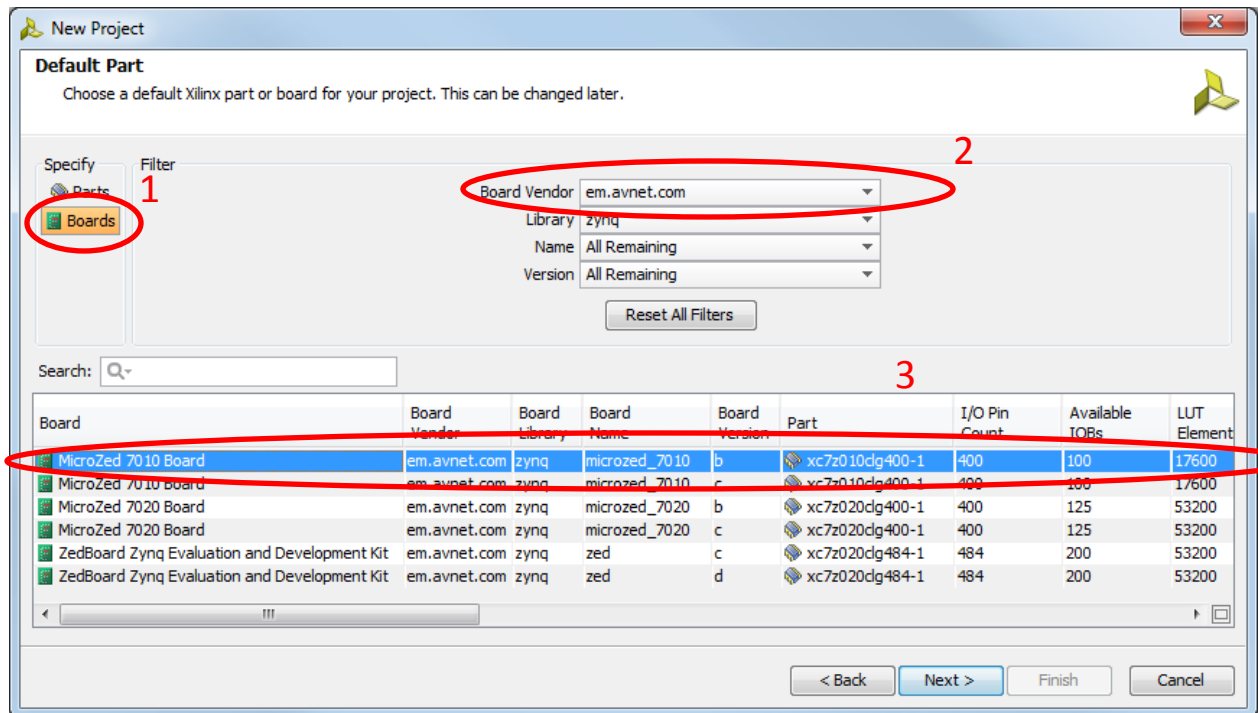


Figure 6 – Select the Target Board = MicroZed 7010

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

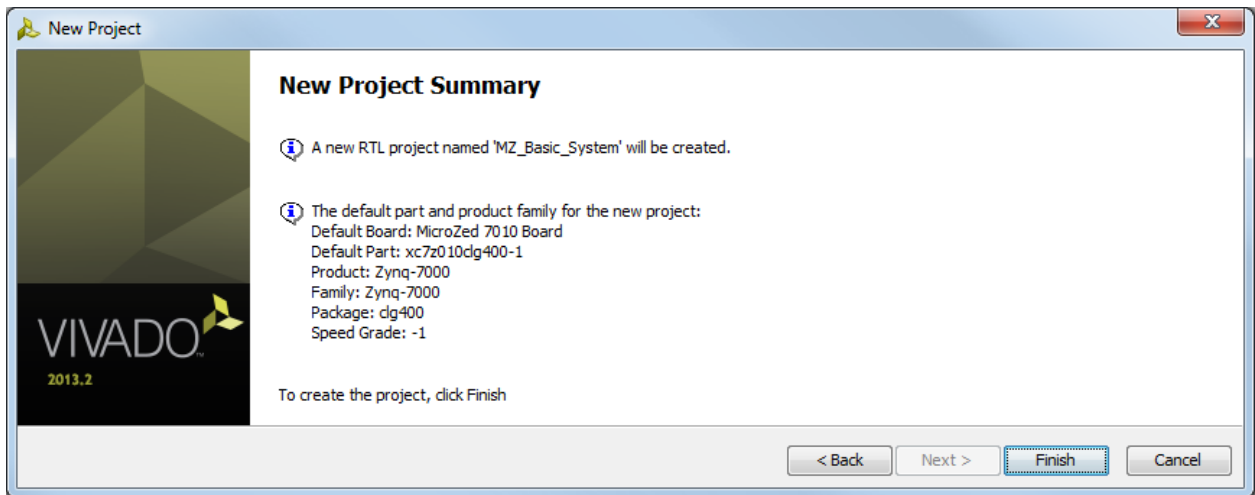


Figure 7 – New Project Summary

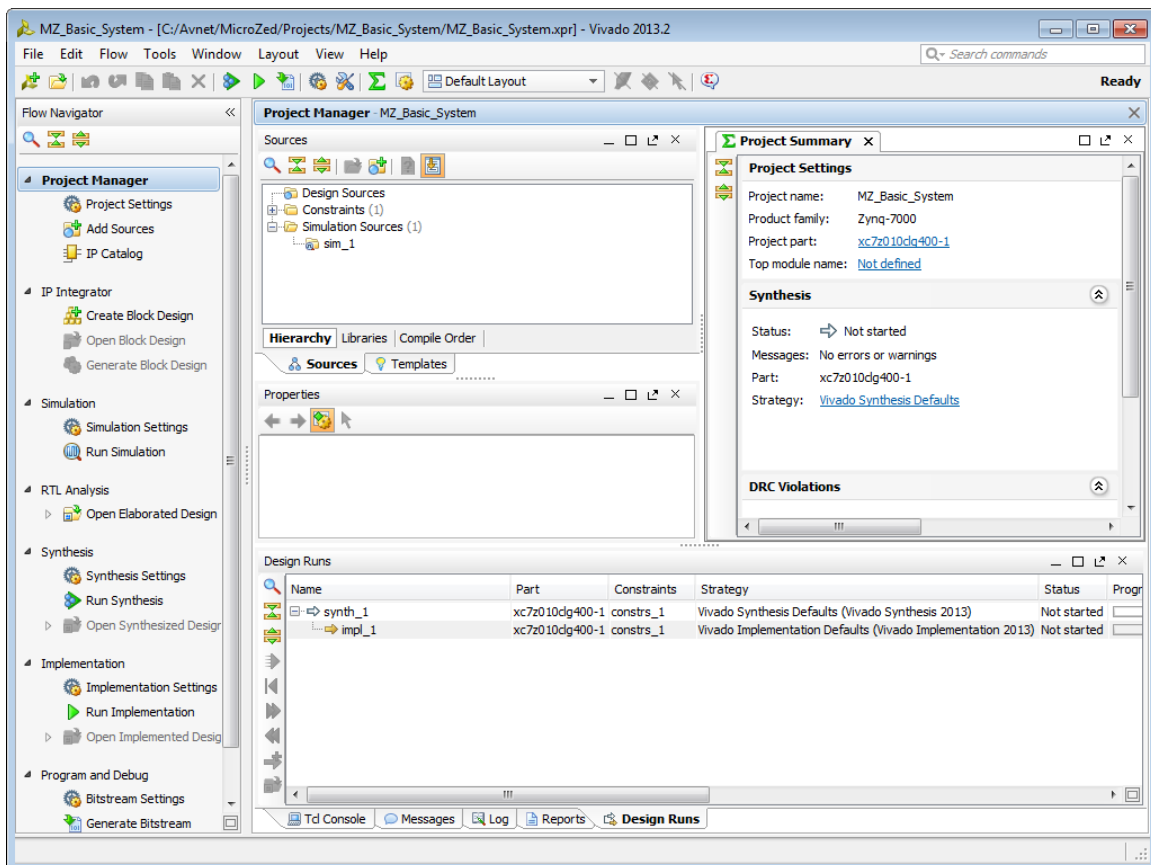


Figure 8 – Vivado Cockpit

## Experiment 3: 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**.

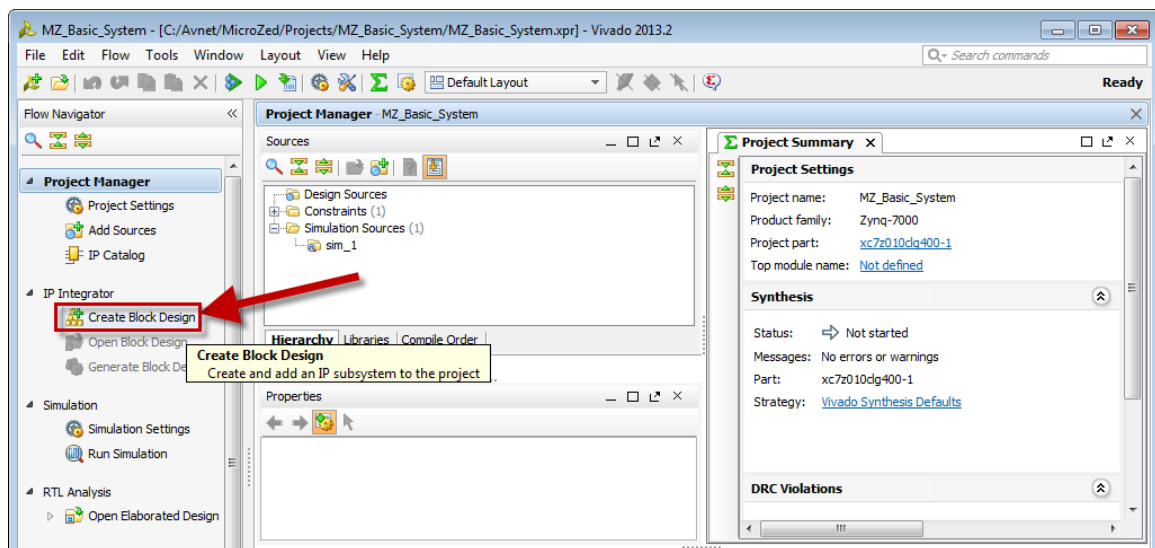


Figure 9 - Create Block Design

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

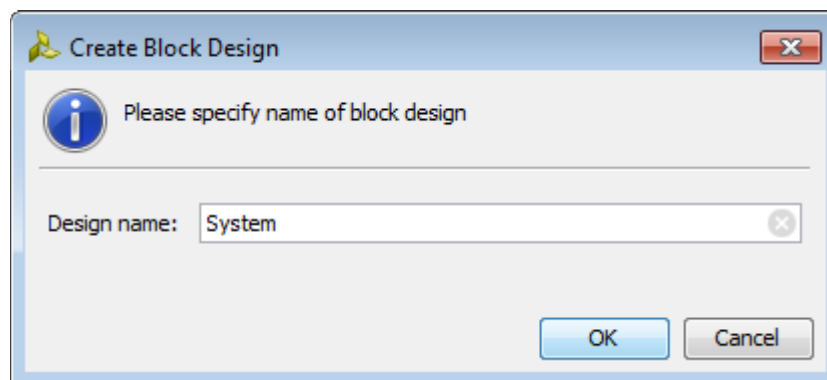


Figure 10 - Block Design Name



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

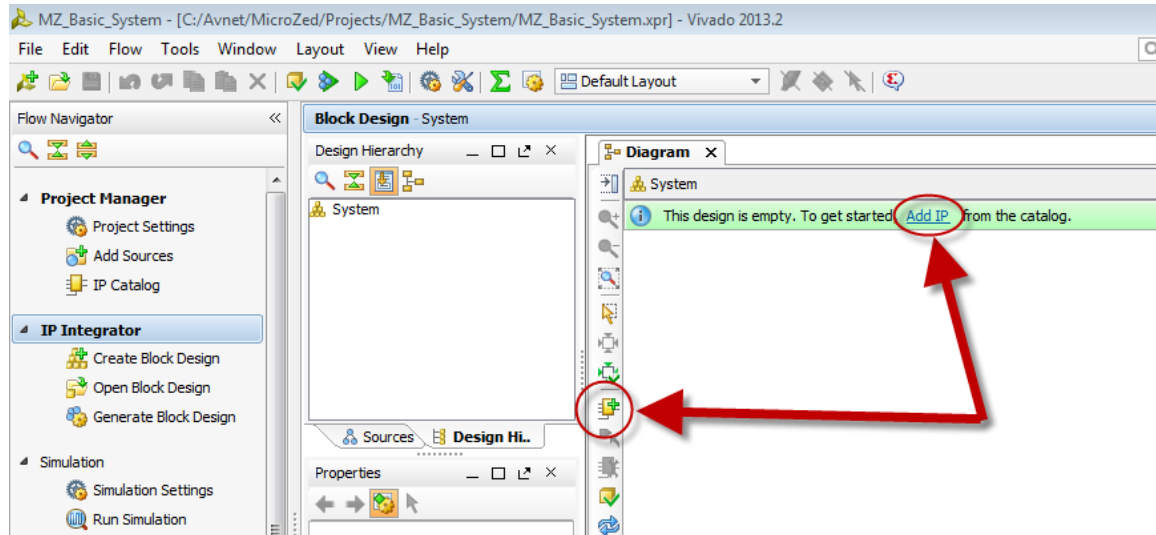


Figure 11 – Add IP to the Block Design

4. The *Add Sources* window opens. Scroll to the bottom of the window. Find the **ZYNQ7 Processing System** IP. Either double-click this or drag and drop to the *Diagram* window.

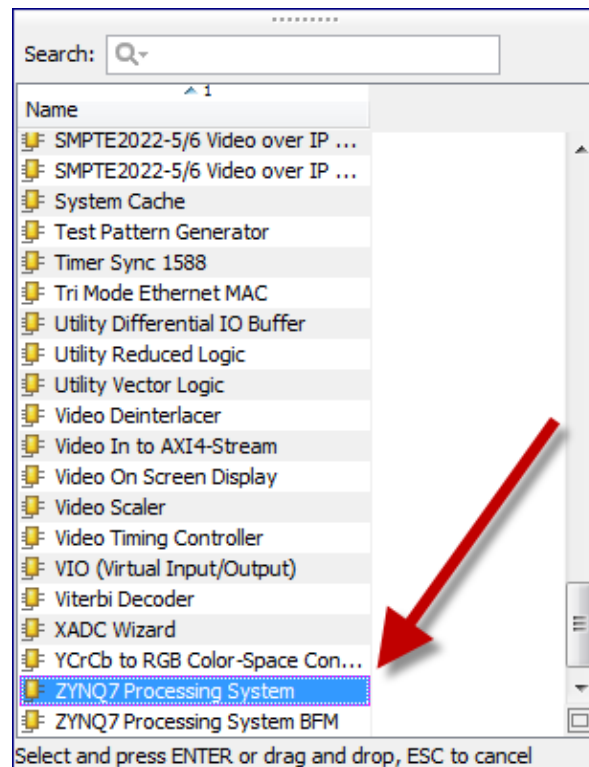


Figure 12 – Add IP Window

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

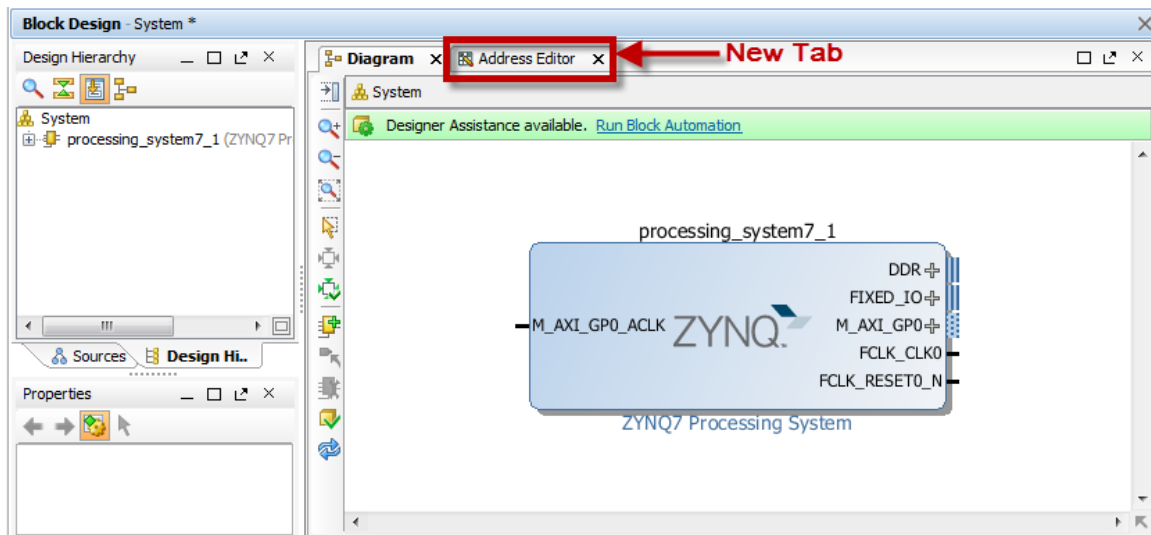



Figure 13 – Updated Block Diagram

One deficiency of Vivado 2013.2 and the 3<sup>rd</sup>-party board awareness is that Vivado does not automatically import the PS configuration. Despite specifying the MicroZed 7010 earlier, the Zynq PS7 has all the tool defaults set at this point. For all designers creating their own custom board, this will be the same situation, and the designer will need to manually configure the entire PS, starting from blank. We could do that as well, but it is a bit time consuming. In the interest of time, we will use a TCL Preset file to import the MicroZed Zynq PS parameters.

5. Download `MicroZed_PS_properties_v01.tcl` from [www.microzed.org](http://www.microzed.org). Copy this file to an easy location to find (like `C:\Avnet\MicroZed`).
6. At the bottom of the Vivado cockpit, select the *Tcl Console* tab.
7. Type `'cd C:/Avnet/MicroZed'` (or insert the directory where you copied the preset file). Note that the TCL Console requires forward slashes, not backward slashes. Click **<Enter>**
8. Type `'source MicroZed_PS_properties_v01.tcl'` and click **<Enter>**

9. To now view the Zynq PS configuration that was created by sourcing the TCL script, do one of three things: 1) double-click the Zynq box; 2) right-click and select **Customize Block**; 3) select the Zynq box and click the  icon. This opens the Zynq Block Design, formally known as the PS Configuration Tool in XPS.

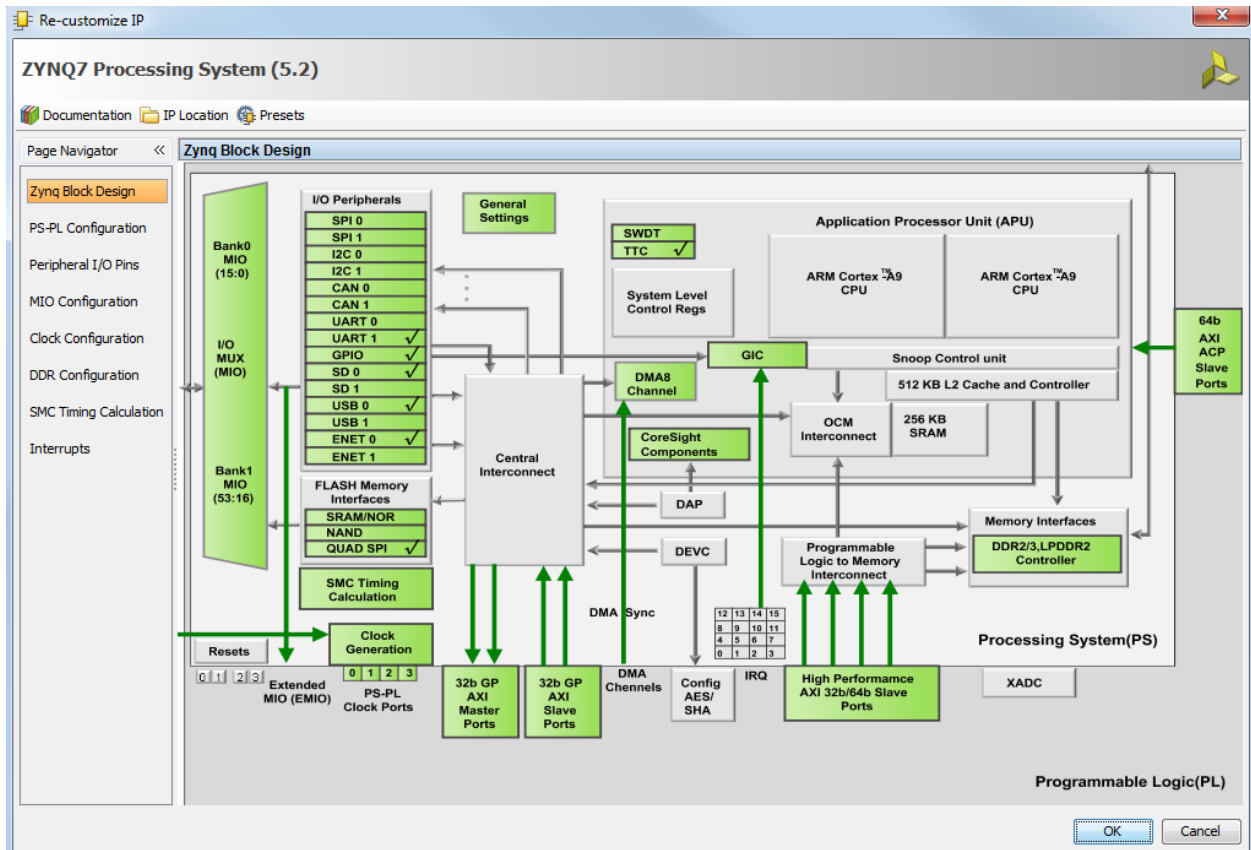


Figure 14 – Zynq Block Design with Configuration Completed

10. Once the file is imported, the PS subsystem may be examined. Explore the predefined MIO peripherals, clock rates, and Memory Controller settings by clicking on any of the green boxes or selecting the listed items under *Page Navigator*.

When done, click **OK**.

11. Back in the Vivado Block Design, click the **Run Block Automation** link at the top of the window and select **/processing\_system7\_1** to connect all block I/O.

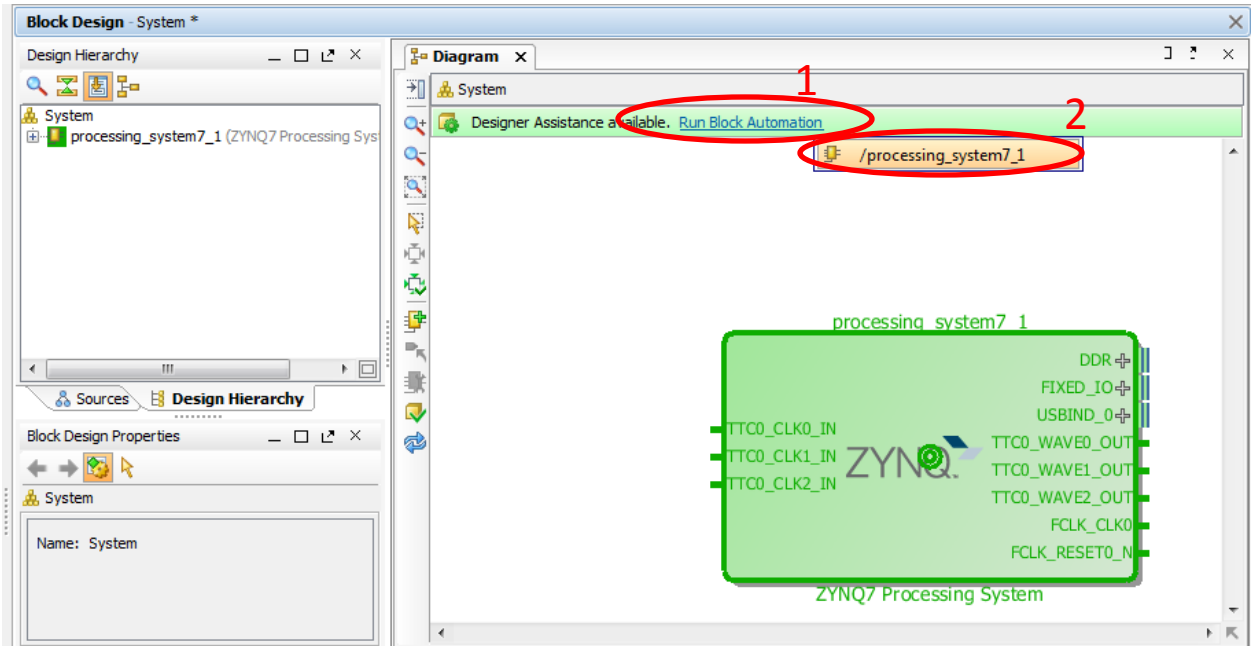


Figure 15 - Run Block Automation

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



Figure 16 – Run Block Automation

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

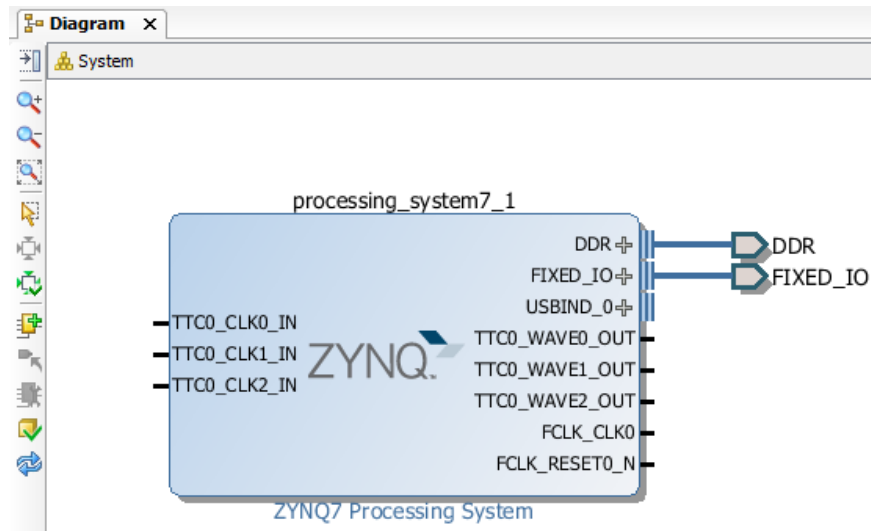



Figure 17 - Zynq Block Diagram with External I/O

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

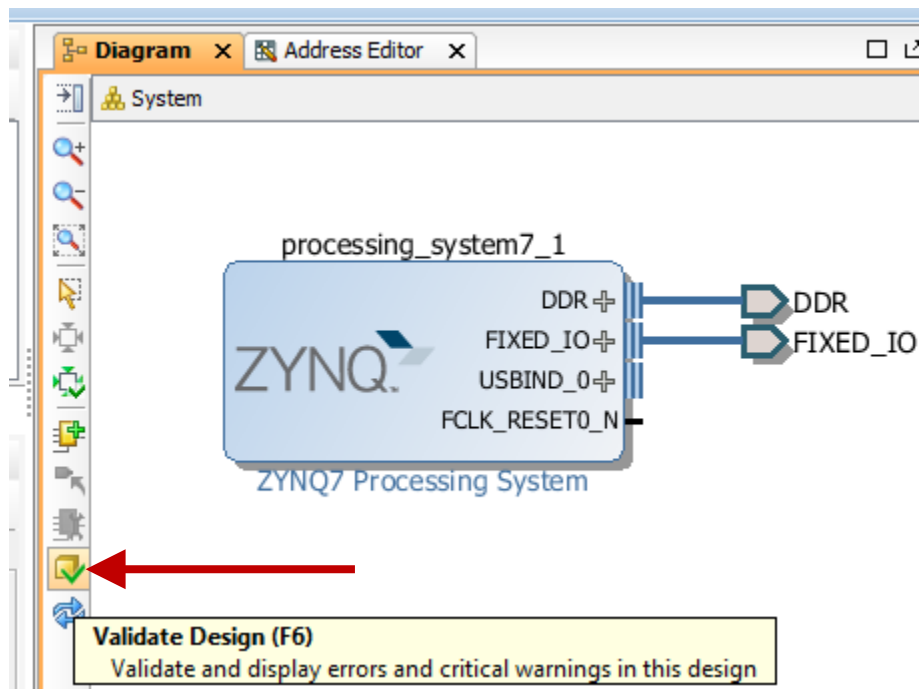


Figure 18 - Validate Zynq Block Design

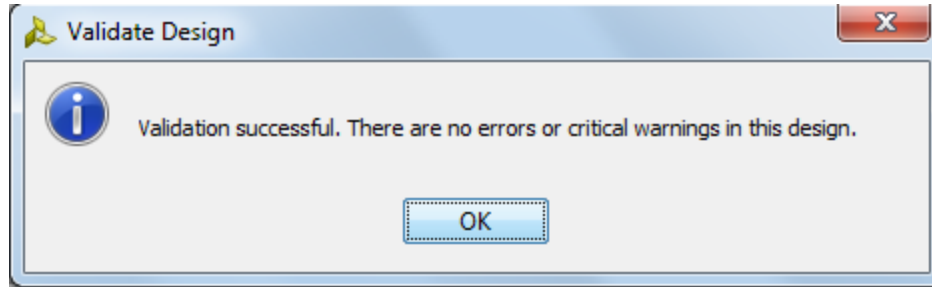
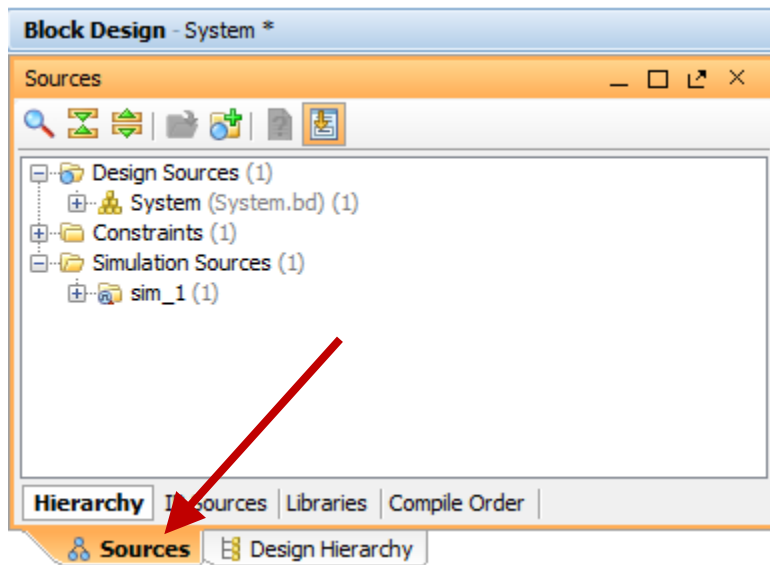


Figure 19 – Validation Successful

15. 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.



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

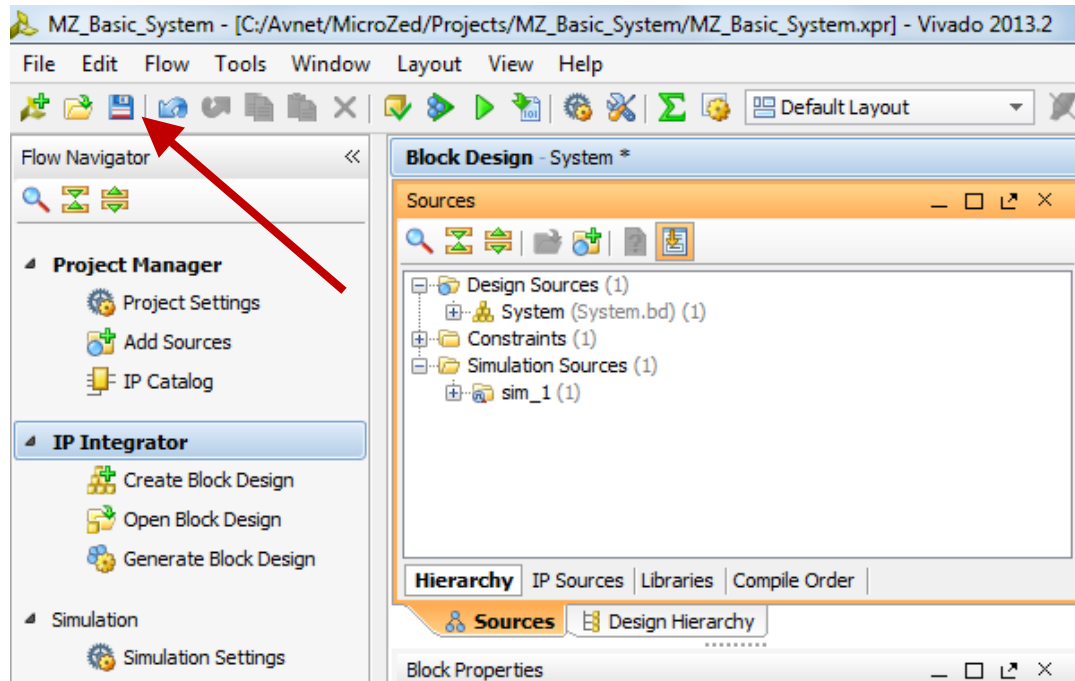


Figure 20 - Sources Tab

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

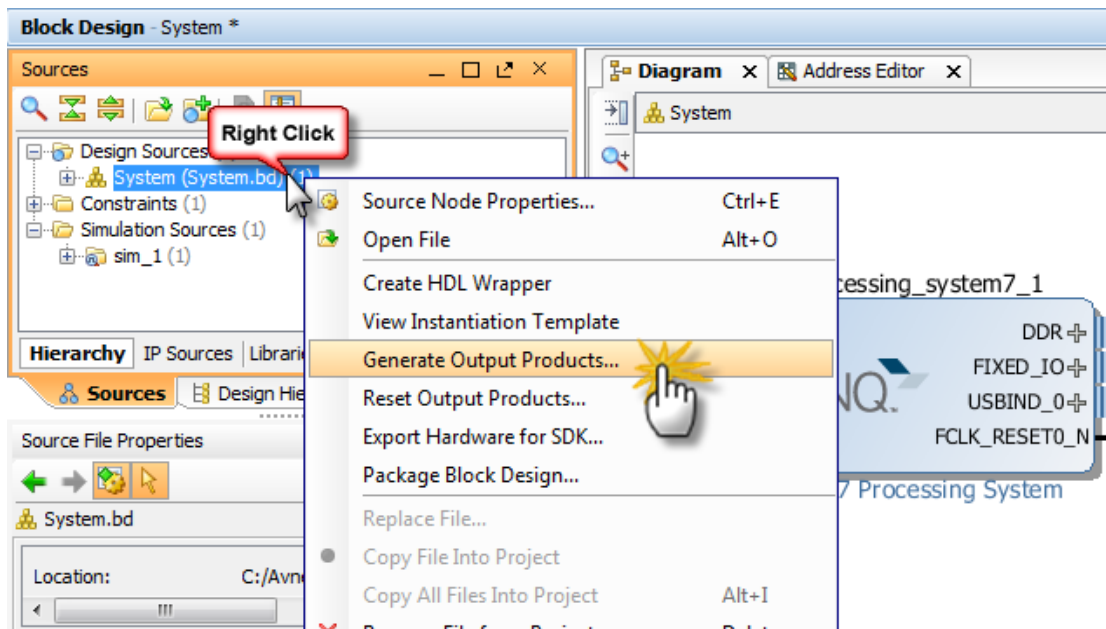


Figure 21 - Generate Output Products

In the Manage Output Products window, you can see the current state of all flows. Contrary to what is shown here, it does not actually run through implementation, it just creates the necessary files for Synthesis and Implementation, similar to the Create Netlist function from XPS.

18. Click **OK**

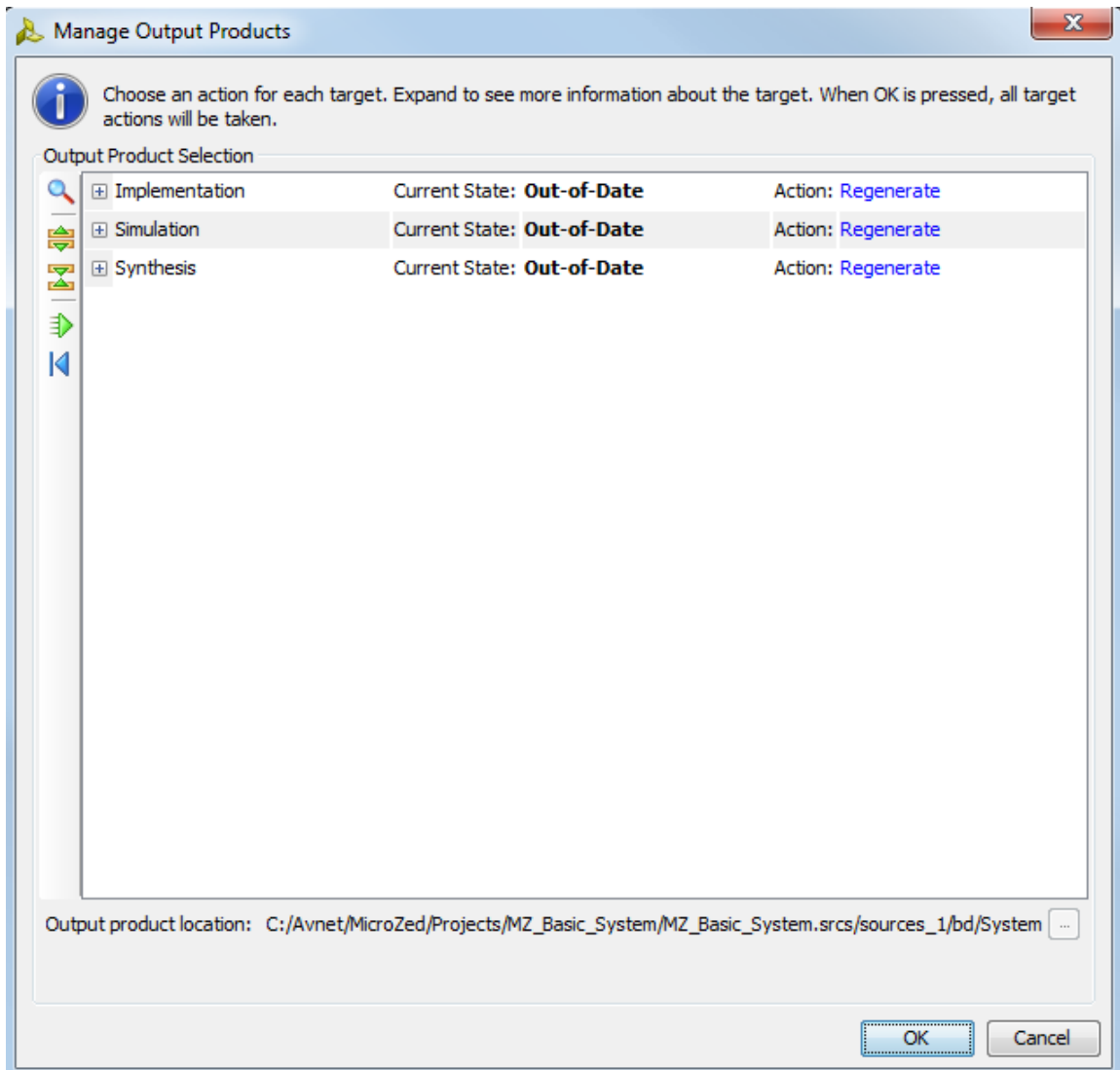


Figure 22 – Manage Output Products



19. Once completed right-click on **System.bd** again and select **Create HDL wrapper**. Once generated, click **OK** to close the information window.

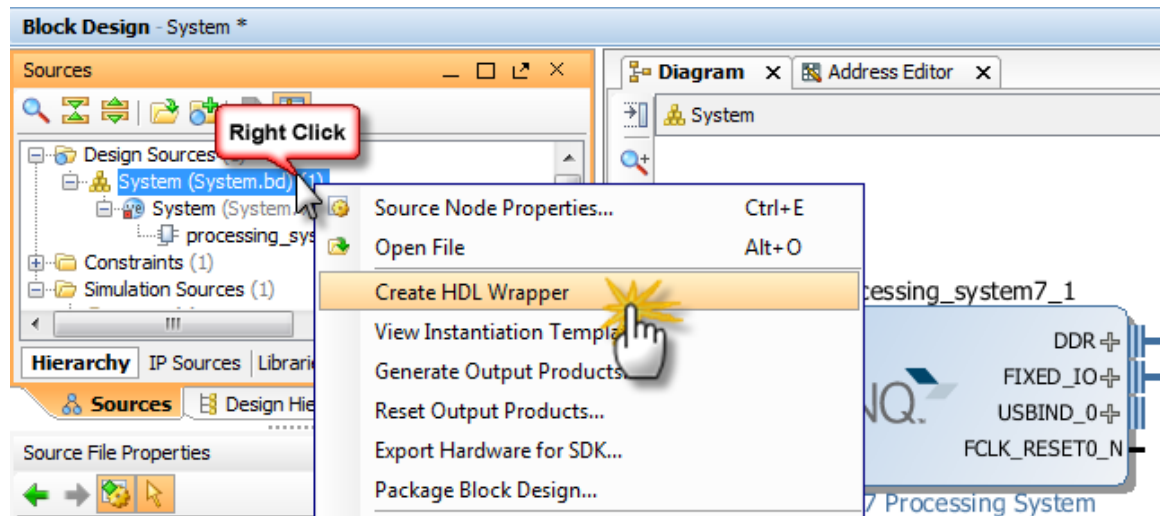


Figure 23 - Create Top Level HDL Wrapper

20. Once the top-level wrapper is created, select it in the *Sources* window then click **Generate Bitstream** in the *Flow Navigator* window (even though this design will not contain a bitstream). Click **Yes** to start Synthesis and Implementation flows. *Check the upper right-hand corner of the tool for a status bar.*
21. When bitstream generation is completed, click **Cancel** to close the info dialog.

## Experiment 4: 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**.

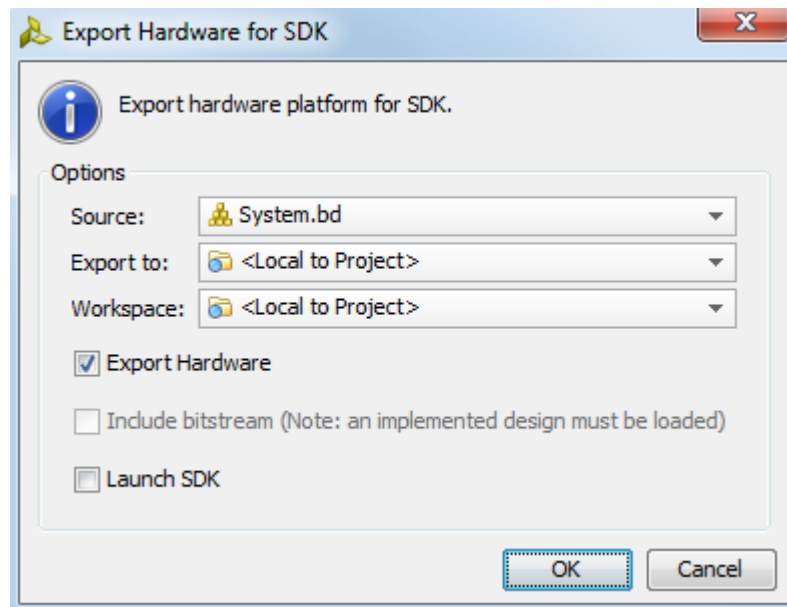


Figure 24 – Export Zynq hardware platform to SDK

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

Name	Date modified	Type	Size
MZ_Basic_System.data	8/22/2013 5:10 PM	File folder	
MZ_Basic_System.runs	8/22/2013 5:13 PM	File folder	
MZ_Basic_System.sdk	8/22/2013 5:21 PM	File folder	
MZ_Basic_System.srscs	8/22/2013 4:49 PM	File folder	
MZ_Basic_System.xpr	8/22/2013 5:10 PM	Vivado Project File	1 KB

**Figure 25 – 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 .data, .runs, and .srscs 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.

Name	Date modified	Type	Size
ps7_init.c	8/22/2013 5:22 PM	UltraEdit Docume...	537 KB
ps7_init.h	8/22/2013 5:22 PM	UltraEdit Docume...	6 KB
ps7_init.html	8/22/2013 5:22 PM	Firefox HTML Doc...	648 KB
ps7_init.tcl	8/22/2013 5:22 PM	TCL File	34 KB
ps7_summary.html	8/22/2013 5:21 PM	Firefox HTML Doc...	0 KB
System.xml	8/22/2013 5:22 PM	XML Document	225 KB

**Figure 26 – Zynq Hardware Platform Export for SDK**

---

## Revision History

Date	Version	Revision
23 Aug 2013	2013_2.01	Initial Avnet release for Vivado 2013.2