Exploring the Architecture of the Zynq-7000 All Programmable SoC

2016.3

Abstract

This introduction to the basic process of instantiating and customizing the processor system (PS) of the Zynq®-7000 All Programmable SoC family of parts illustrates the process of customizing the PS. While not every aspect of customization is covered, the processes provided here can be extended to all aspects of customization.

This lab should take approximately 20 minutes.

Objectives

After completing this lab, you will be able to:

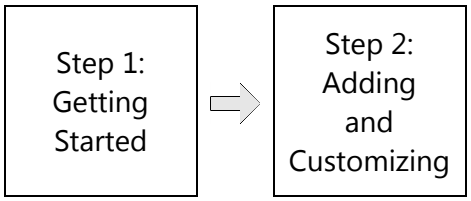
- Instantiate a processor system
- Customize a processor system
- Run the Block Automation tool for a processor system

Introduction

This lab introduces you to the basic process of managing the processing system block in the Zynq-7000 All Programmable SoC family of devices. Included in management is the instantiation of the PS, re-customization, and finally running the Block Automation tool.

The PS is a complex subsystem and an exhaustive investigation of every aspect of this subsystem is beyond the scope of this lab. Only some of the more popular customization are illustrated here with the hope that you will gain familiarity with how these options are organized and that you can quickly find whatever aspect of customization you desire to modify.

General Flow



Getting Started Step 1

As this is a hardware lab focusing on how the Cortex-A9 processor of the Zynq-7000 AP SoC can be configured, all your work will be done with the Vivado Design Suite IP integrator (IPI).

You will begin by launching the Vivado Design Suite, then loading a helper Tcl script to properly configure the project and quickly get you to the IPI tool. If you need to refresh your memory regarding the details of project creation or the mechanics of using the IPI tool, refer to the "Driving the IPI Tool" topic cluster.

There are a number of ways to launch the Vivado Design Suite. The two most popular mechanisms are shown here.

1-1. Launch the Vivado Design Suite.

This can be done in two standard ways, use your preferred method.

1-1-1. Select **Start** > **All Programs** > **Xilinx Design Tools** > **Vivado 2016.3** > **Vivado 2016.3**.

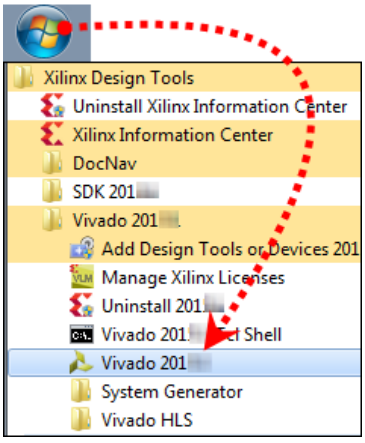


Figure 7-1: Launching the Vivado Design Suite from the Start Menu

-- OR --

Double-click the **Vivado Design Suite** shortcut icon () on the desktop.

The Vivado Design Suite opens to the Welcome window. From the Welcome window you can create a new project, open an existing project, or enter Tcl commands directly into the Vivado Design Suite as well as access documentation and examples.

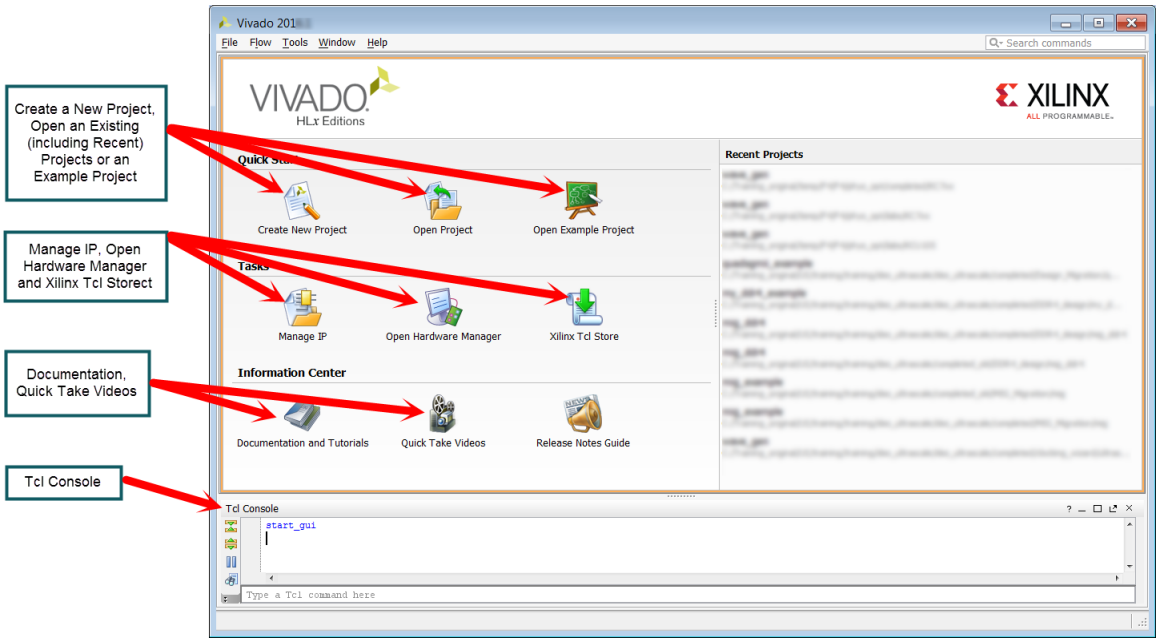


Figure 7-2: Vivado Design Suite Welcome Screen

With the Vivado Design Suite now open, you will load a helper Tcl script and run it to configure the project and get you to the important part of this lab—working with the processor.

The Vivado Design Suite offers both GUI and scripted control. Scripted control takes the form of Tcl commands. These Tcl commands can be entered directly into the tool one at a time, or an entire Tcl script can be loaded and executed.

1-2. Run a Tcl script.

1-2-1. Locate the Tcl command line entry.

The command line entry can be found either on the Welcome page prior to a project being opened, or once a project has been opened.

From the Welcome screen:

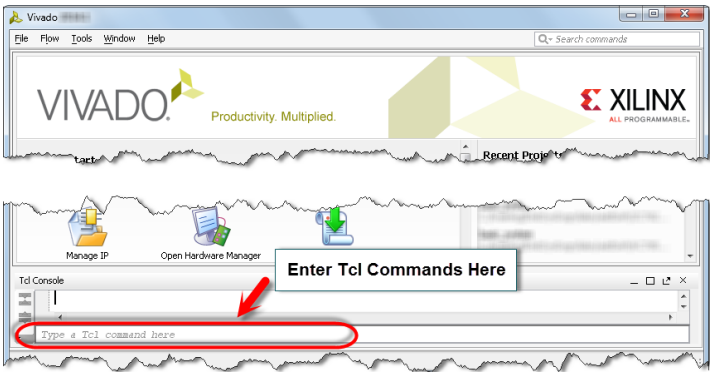


Figure 7-3: Accessing the Tcl Console from the Getting Started Page

From an opened project:

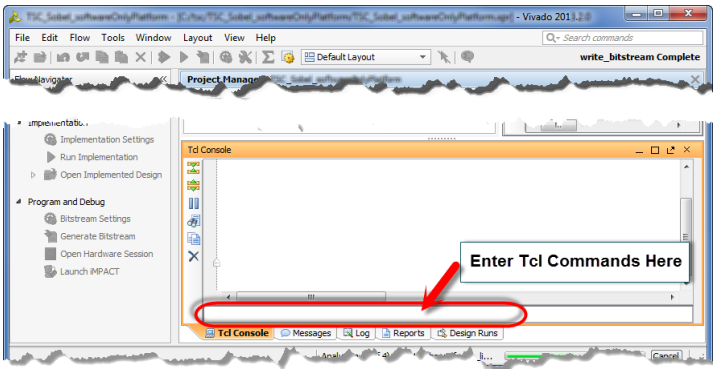


Figure 7-4: Entering Commands into the Tcl Console from an Open Project

The default directory for the Tcl environment is nested within the Xilinx installation directory. This placement, however, is often disadvantageous. In most cases, you will want to navigate to a more useful path. To do this, use the `cd` command to change directory to the user directory.

1-2-2. Change the current working directory to where the Tcl script is located by entering:

```
cd C:\training\ArchZynq7000_Overview\support
```

Remember that the Tcl environment is based on Linux and requires the '/' character to delimit hierarchical paths.

1-2-3. Verify that you are now where you want to be by entering the following into the Tcl command line:

```
pwd
```

The current working directory is displayed. If you are not where you want to be, use the `cd` command to change to `C:\training\ArchZynq7000_Overview\support`.

1-2-4. Enter the following Tcl command:

```
source exploreZynq7000_completer.tcl
```

The Tcl script is run as though you typed each command included in the Tcl script into the Tcl command line. You can follow the execution of the script and monitor for any errors or warnings in the Tcl Console.

With the Tcl script now loaded, you will run the *createProject* and *createBlockDesign* procs to build the Vivado Design Suite project and create an appropriately named block design in which you will perform the remainder of this lab.

1-3. Run the *createProject* and *createBlockDesign* procs.

1-3-1. Enter the following into the Tcl command line to create the Vivado Design Suite project:

```
createProject
```

1-3-2. Enter the following into the Tcl command line to create the block design in which you will build your processor system:

```
createBlockDesign "Zynq7KembdDsgn"
```

Adding and Customizing the Zynq-7000 AP SoC Processor Step 2

With the block diagram canvas open, you will now add and customize a Zynq-7000 AP SoC processor.

2-1. Add a ZYNQ7 Processing System to the IP Integrator canvas.

If you do not recall how to perform this task, refer to the "Adding a Processor to the Block Design" section under IP Integrator Operations in the *Lab Reference Guide*.

Remember that adding a processor IP is just like adding any other type of IP to the canvas. You can also review the "Driving the IPI Tool" topic cluster for a guided example.

Question 1

What information does the wizard provide?

Question 2

How does configuring the PS of the Zynq-7000 AP SoC differ from configuring a MicroBlaze processor?

The Zynq All Programmable PS contains a large number of customizable features. The following instructions will illustrate how to access various sections of the Zynq All Programmable PS, not necessarily indicating which settings to configure.

2-2. Access the Re-customization dialog box.

2-2-1. Double-click the **Zynq PS** icon.

-- OR --

Right-click the **Zynq PS** icon and select **Customize Block**.

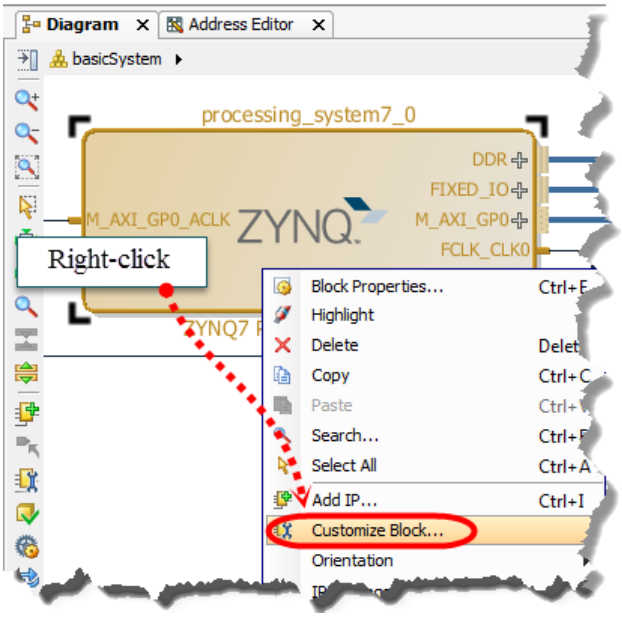


Figure 7-5: Opening the Re-customization Dialog Box for the Zynq All Programmable SoC PS

The Re-customization dialog box opens to reveal the Zynq block design.

2-3. Import board-specific settings.

If you are using an evaluation board, you may want to load the settings for this board as it will contain many of the parameters specific to that board (such as DDR memory timing parameters, enabled peripherals supported by that board, etc.)

You can also create a *.bd* file for your custom board if you want and import those settings.

2-3-1. Click **Presets**.

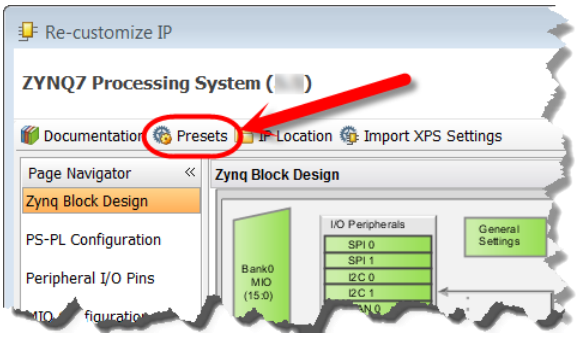


Figure 7-6: Accessing the Presets Button

2-3-2. Select the template for **ZedBoard**.

All the Xilinx boards that support the device that was selected during the project creation are shown under the System Template (Presets) drop-down list. Even if you selected the part by board, you have the opportunity to select a different board that has the same part on it.

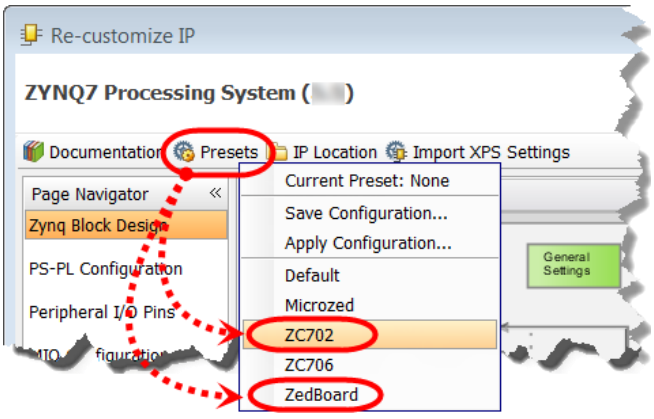


Figure 7-7: Selecting a Preset Template (ZedBoard and ZC702 Shown as Examples)

The next instruction provides you with general guidance as to how to customize various aspects of the PS. At the end of this instruction you will be provided with specific guidance regarding how you are to customize the PS.

2-4. Select the page or specific block to re-customize.

2-4-1. Click the topic in the Page Navigator or any green (active) box from the Zynq Block Diagram.

Along the left are the navigational tabs that you will use to access different groups of parameters.

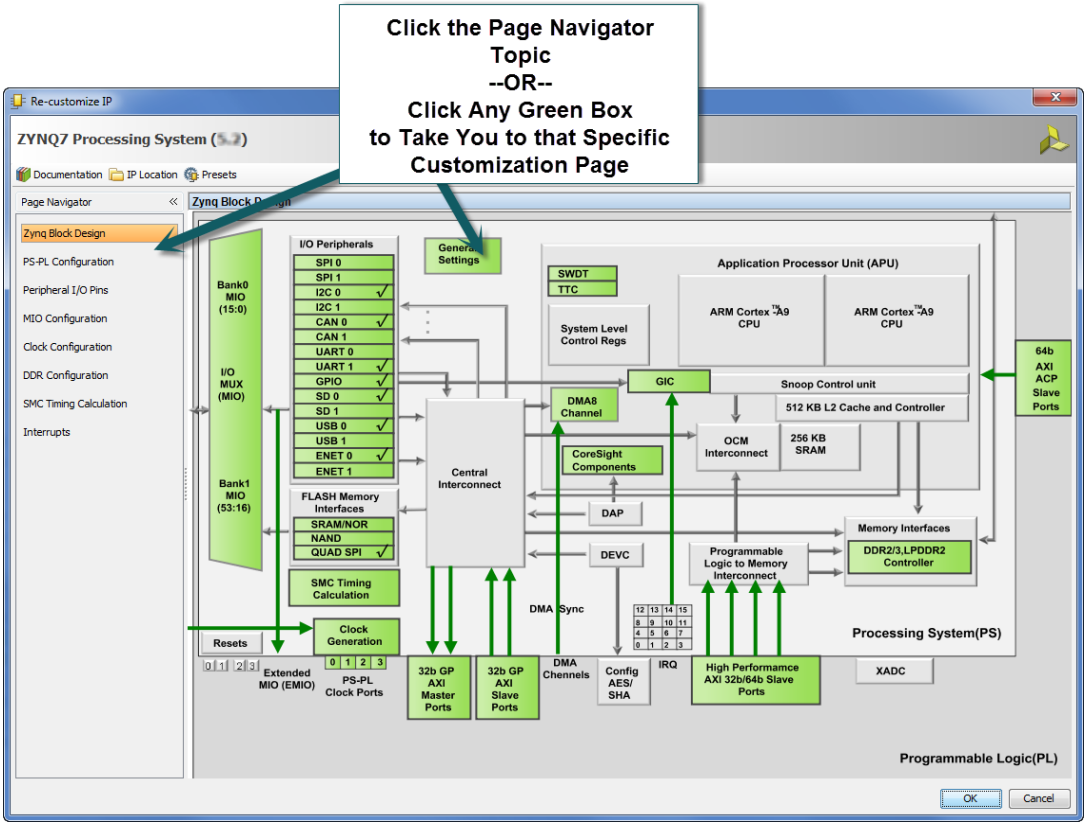


Figure 7-8: Navigating the Zynq PS Recustomization Dialog Box

- PS-PL Configuration: This page contains all the signals that cross the PS-PL boundary. These signals are broken down further into:
  - General: Contains default baud rates for the UARTs, FTM Trace buffer settings, PS-PL cross triggering enables, enables for the clock triggers, and reset.
  - DMA Controller: Contains enables for the four peripheral request interfaces.
  - GP Master AXI Interface: Enables/disables access to the GP Master AXI Interfaces to the PL.
  - GP Slave AXI Interface: Enables/disables access to the GP Slave AXI Interfaces from the PL.
  - HP Slave AXI Interface: Enables/disables access to the GP Slave AXI Interfaces from the PL and sets the port width.
  - ACP Slave AXI Interface: Enables/disables access to the ACP Slave AXI Interface from the PL.

- Peripheral I/O Pins
- MIO Configuration
- Clock Configuration
- DDR Configuration
- SMC Timing Calculation
- Interrupts

The following is how your system should be configured:

- Use the ZC702 or Zed preset
- Disable the following features:
  - All peripherals except as listed below (accessible through either the Peripheral I/O Pins or the MIO configuration or Block Diagram > I/O Peripherals)
    - UART 1
    - QSPI
- Enable the following features:
  - HP Slave AXI Interface > S_AXI_HP0 interface
  - QSPI (enabled by default through the preset)
  - UART1 (enabled by default through the preset)

2-4-2. Configure the PS according to the list above.

2-4-3. Click **OK** to accept the configuration and return to the IPI canvas.

Remember that you can re-configure the processor at any time. However, use caution as you may need to re-run Designer Assistance to handle any changes you make.

Many IP blocks are supported by the Designer Assistance feature for automating the configuration of an IP block as well as block-specific connections. This is referred to as Block Automation and allows designers to quickly configure new IP blocks for common use cases.

2-5. Use Block Automation to automate the configuration of the recently instantiated IP.

2-5-1. Click **Run Block Automation** from the Designer Assistance information bar.

Figure 7-9: Designer Assistance Offering Block Automation

This opens a Run Block Automation dialog box listing all the IP currently in the design eligible for block automation. IPs are listed in a hierarchy on the left and any options associated with a particular automation will be shown in the right pane whenever an IP instance is selected in the left pane.

2-5-2. Click the **Expand All** icon ( ) to ensure that the list of available automations is fully visible.

2-5-3. Select either the top-level **All Automation** check box or individual blocks as listed below. Unless otherwise indicated within the block list, maintain default automation options for all blocks.

- Blocks: processing_system7_0

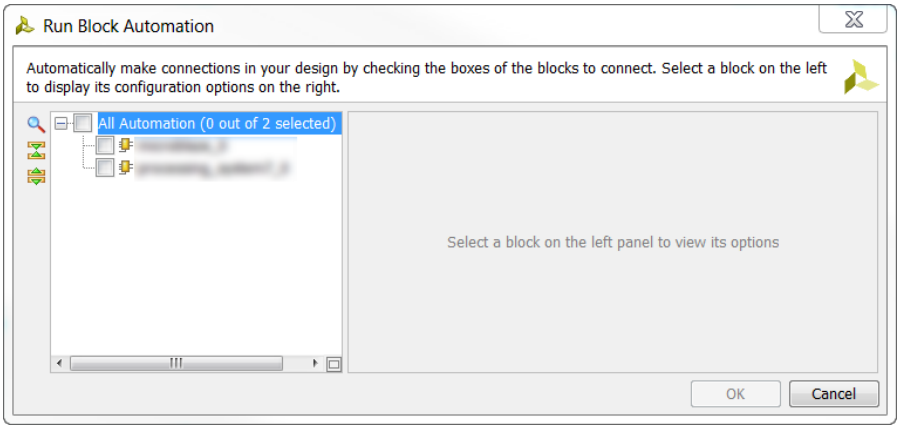


Figure 7-10: Run Block Automation Dialog Box

Note the various options that are selectable for the PS.

Question 3

Do any of these options look familiar?

2-5-4. Deselect the **Apply Board Preset** option to avoid overwriting the current PS settings.

The Cross Trigger In and Out selections are used in co-debugging to enable the processor(s) to cause the debug logic (ILAs) in the PL to trigger and to have the trigger signal generated by the debug logic (ILAs) cause the software to pause (a la breakpoint).

2-5-5. Click **OK** to run the selected automation.

Question 4

What does the Block Automation do?

At this point, the PS has connections to external memory (DDR), has its own internal memory for booting, and has a collection of peripherals available for I/O. As is, this could operate with no further additions.

Question 5

What happens to the unconnected port to the processing system?

Summary

You just completed the basic tasks of instantiating the processing system of a Zynq-7000 AP SoC, configuring it, and using the Block Automation tool to connect the DDR and fixed I/O to the outside world.

Answers

1. What information does the wizard provide?

   The opening page of the wizard provides two methods for locating the aspect of the processing system that can be modified. The Page Navigator (on the left side of the wizard) provides a textual list of the various configurable areas, whereas the Zynq Block Design graphic enables you to quickly locate a configurable aspect by clicking any of the green blocks.

   Along the top menu, you can retrieve documentation for this processor, load a preset (pre-configurations for various boards including DDR settings, Flash configuration, etc.), and other less frequently used options.

2. How does configuring the PS of the Zynq-7000 AP SoC differ from configuring a MicroBlaze processor?

   The MicroBlaze processor is a softcore processor and the various configurations will impact overall performance, the number and type of resources used, and maximum clock frequency.

   The Zynq AP SoC's processing system is dedicated silicon and, while peripherals, clocks, and some connections can be configured, the maximum frequency, performance, and consumed resources is fixed and cannot be changed.

3. Do any of these options look familiar?

   The Apply Board Preset option is a duplicate of the process that you performed inside the Re-customization window. If you leave the Apply Board Preset option selected at this stage, it will re-configure the PS to the default board settings and you will have to re-customize the PS again.

4. What does the Block Automation do?

   The Block Automation makes the connections to the DDR and the fixed I/O. These are the dedicated pins of the PS. Because they are dedicated, they do not require any constraints.

   Remember that the fixed I/O is controlled via the MUI and can be dynamically reconfigured.

5. What happens to the unconnected port to the processing system?

   The unused ports (even though they were defined) connect between the PS and PL. If there is no logic in the PL, then these ports simply *dangle* and are not used.

   One should be careful, though, not to drive AXI transactions out the master AXI general-purpose port because, since there is no *receiving* AXI logic, the port will hang and likely crash the system.