Developing Zynq Software with Xilinx SDK

Lab 2

Importing the Hardware Platform into SDK

November 2013
Version 03

Lab 2 Overview

The Xilinx® Software Development Kit (SDK) is a full-fledged embedded software development environment for Xilinx embedded processors. SDK is based on the Eclipse open source platform. SDK features include the following:

- Feature-rich C/C++ code editor and compilation environment
- Project management
- Application build configuration and automatic Makefile generation
- Error navigation
- Well-integrated environment for source-level debugging and profiling of embedded targets
- Project archiving

In addition to the above native Eclipse provided features, SDK also provides the following tools for use in Xilinx embedded software development:

- Reference software applications
- Xilinx Microprocessor Debugger (XMD): A debug agent used to communicate with Xilinx embedded processors using JTAG. It also provides various low-level debugging features not directly available in SDK.
- SoC programmer: Used to program the Xilinx SoC with the bitstream.
- Flash programmer: Used for burning bitstreams and software application images into external Flash devices
- FSBL generator: Used for automatically bootloading your embedded software applications from Flash
- Linker script generator: Used for quickly mapping your application image across the hardware memory space

Lab 2 Objectives

When you have completed Lab 2, you will know how to do the following:

- Set up an SDK workspace
- Import a pre-built Zynq hardware platform.
- Briefly examine the hardware platform including documentation for each peripheral.

Experiment 1: Workspace and Hardware Platform

This experiment shows how to set up a workspace and import a hardware platform.

Workspaces

SDK uses the concept of workspaces to hold your software development work. A workspace is a directory in your file system that SDK uses to hold meta-information about the projects with which you are working. The workspace also contains your SDK settings, software project files, and logs.

For convenience, SDK can be instructed to remember your last used workspace. When you enable this option, you will not be prompted to choose a workspace every time you start SDK. You can always switch your current workspace within SDK.

IMPORTANT NOTE: It is not safe to copy or move workspaces from one location to another. If you want to copy or move software components in your workspace, you must use the SDK Export/Import functions. This will be covered in Lab 8.

Hardware Platform

As you learned in Lab 1, the Hardware Platform Specification file (XML) captures all the information and files from a Xilinx® Vivado hardware design that is required for a software developer to write, debug, and deploy software applications for that hardware.

Each Workspace will typically have one hardware platform. There may be multiple BSPs and software applications, but these will always be associated with a single hardware platform.

Experiment 1 General Instruction:

Launch SDK. Create a Workspace in the `SDK_Workspace` directory. Import the Zynq hardware platform reviewed during Lab 1.

Experiment 1 Step-by-Step Instructions:

1. Launch SDK by selecting **Start → All Programs → Xilinx Design Tools → Vivado 2013.3 → SDK → Xilinx SDK 2013.3**.

2. If you have the default settings, SDK will ask you which workspace that you would like to open. We will open a new workspace. If you do not get this view, then select **File → Switch Workspace → Other**. Browse to `C:\Speedway\ZynqSW\2013_3\SDK_Workspace` and click **OK** to select the folder, then **OK** again to close the *Workspace Launcher*.

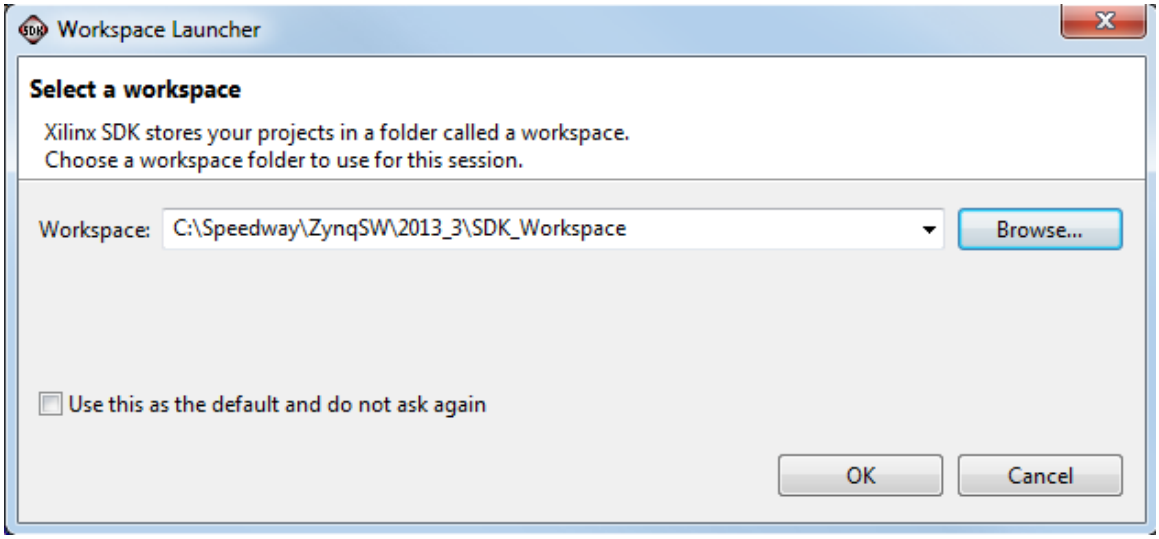

Figure 1 – SDK Workspace, *Spaces Not Allowed*

Next, the Hardware Platform will be imported. The hardware platform for this lab was previously designed and built for the Zynq 7020 SoC on the Avnet ZedBoard. ***This hardware design process is not covered during this software-focused Speedway.*** To learn about this topic, please attend the Avnet Course *Developing Zynq Hardware*.

1. Select **File → New → Other**.

2. Expand the *Xilinx* item, and select **Hardware Platform Specification**. Click **Next >**.

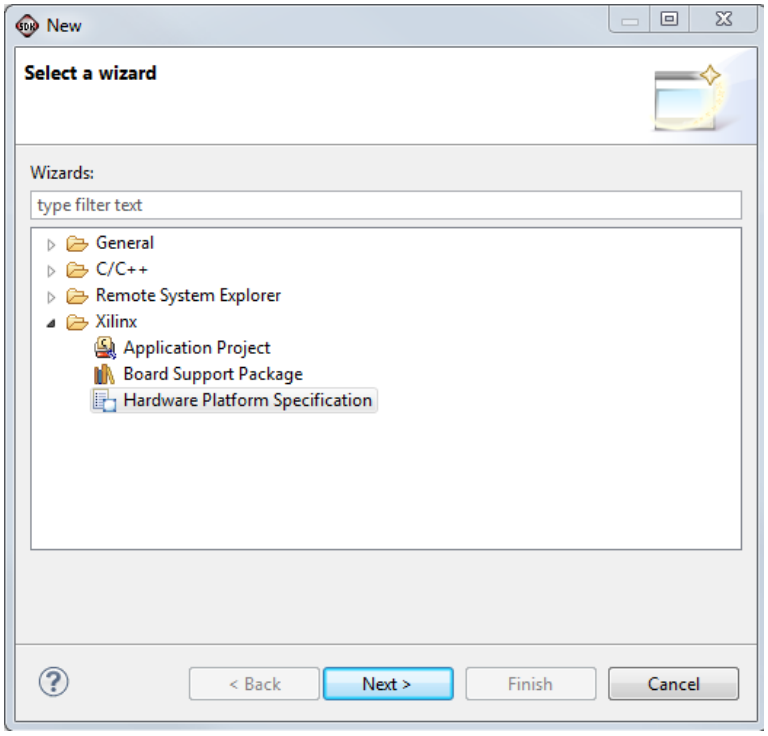

Figure 2 – Creating a New Hardware Platform

3. Insert **ZynqHW** for the *Project name*. Leave the *Use default location* checked.

4. Under *Target Hardware Specification*, click **Browse** and select the `Z_system.xml` file located in the Vivado project folder:

`C:\Speedway\ZynqSW\2013_3\ZynqDesign\ZynqDesign.sdk\SDK\SDK_Export\hw\`

5. Click **Open**. (Do not click Finish yet.)

6. Expand *Bitstream and BMM Files* to see what additional files were identified as part of the hardware platform. Notice that a Bitstream is included, but the BMM is blank. BMM stands for Block RAM Memory Map. If the Block RAMs in the Zynq PL were being initialized as part of this project, then the hardware platform would include a BMM file. Since this hardware platform does not include the BMM file, we can assume that all PL Block RAMs will be uninitialized at boot. Click **Finish**.

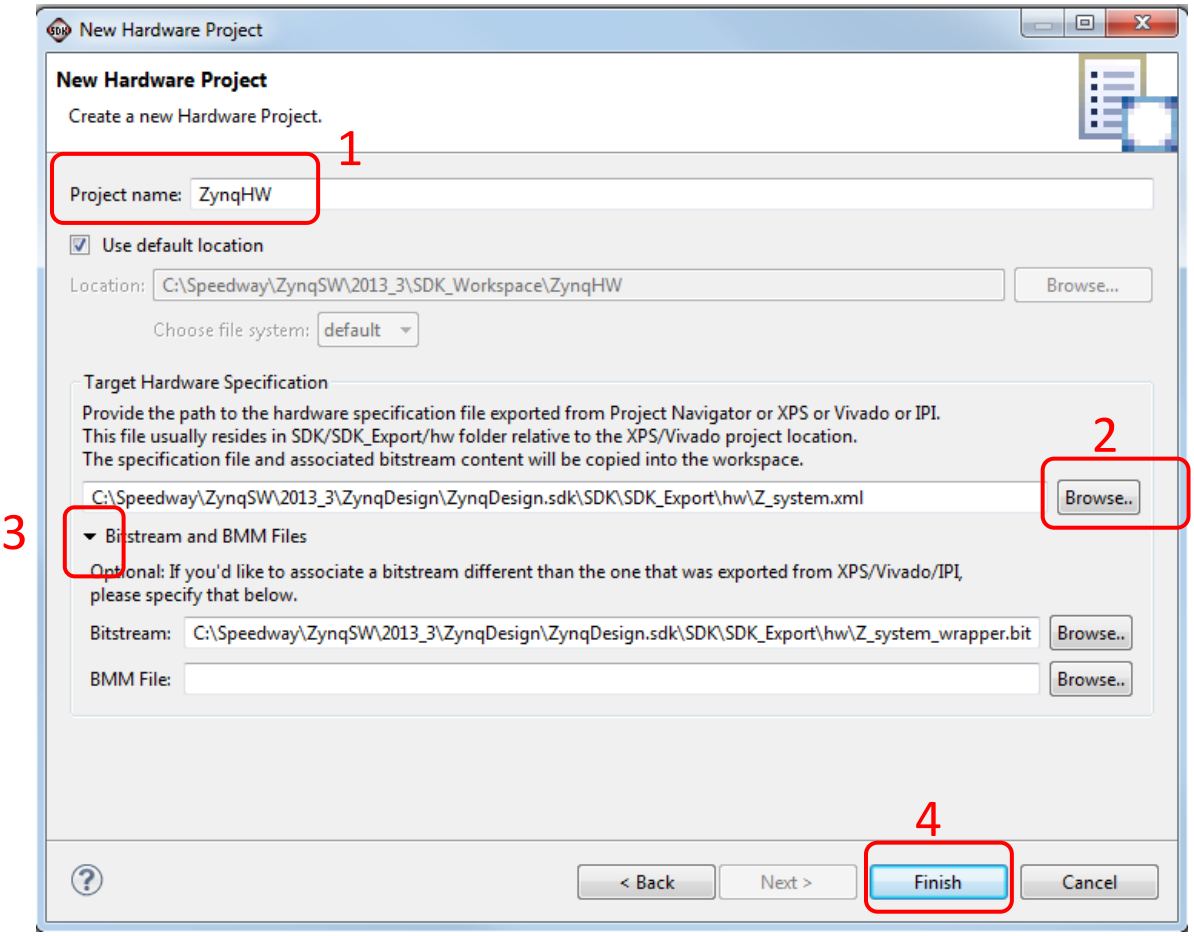

Figure 3 – Import Hardware Platform from Vivado

7. Close the *Welcome* screen if it is still open. Notice the PS7 Zynq hardware platform is now visible in the *Project Explorer*.

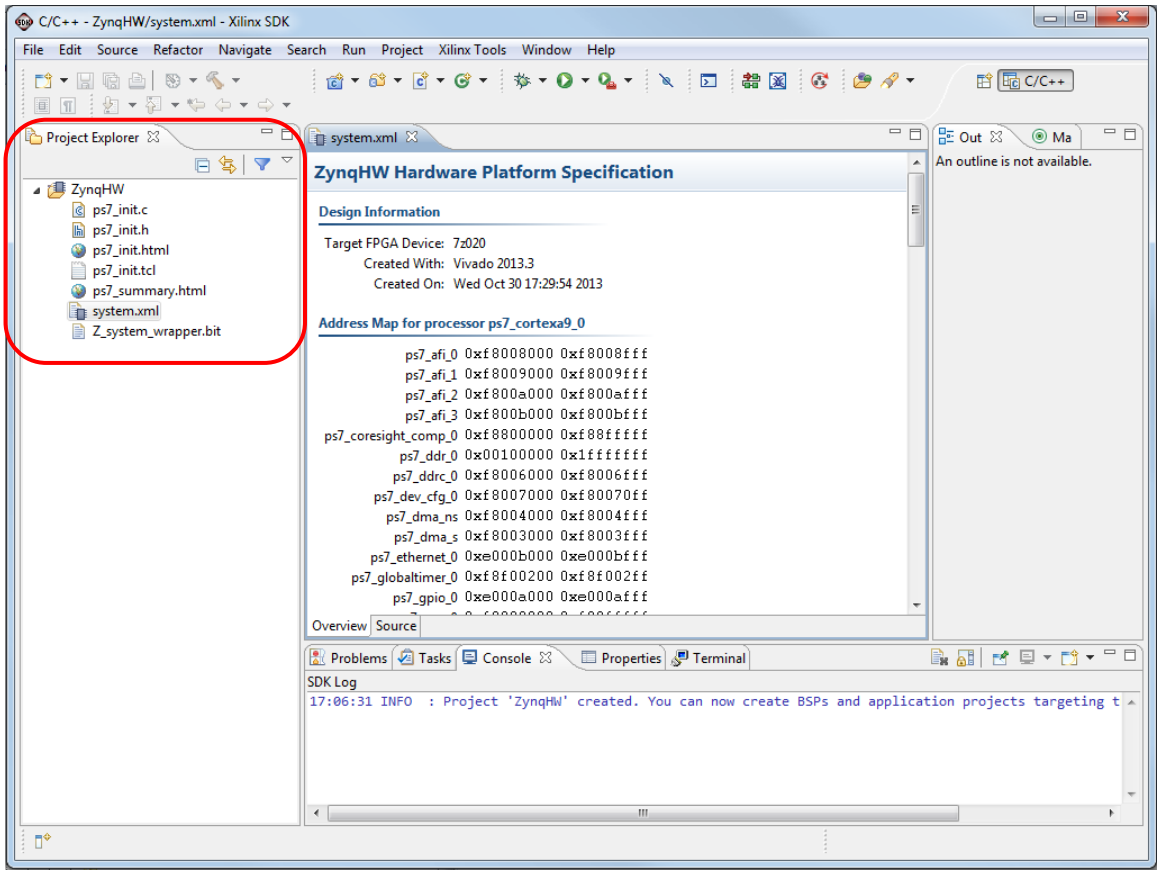

Figure 4 – Hardware Platform Imported and Ready for Use

8. If at any time you see a Windows Security Alert as shown below, click **Allow access**.

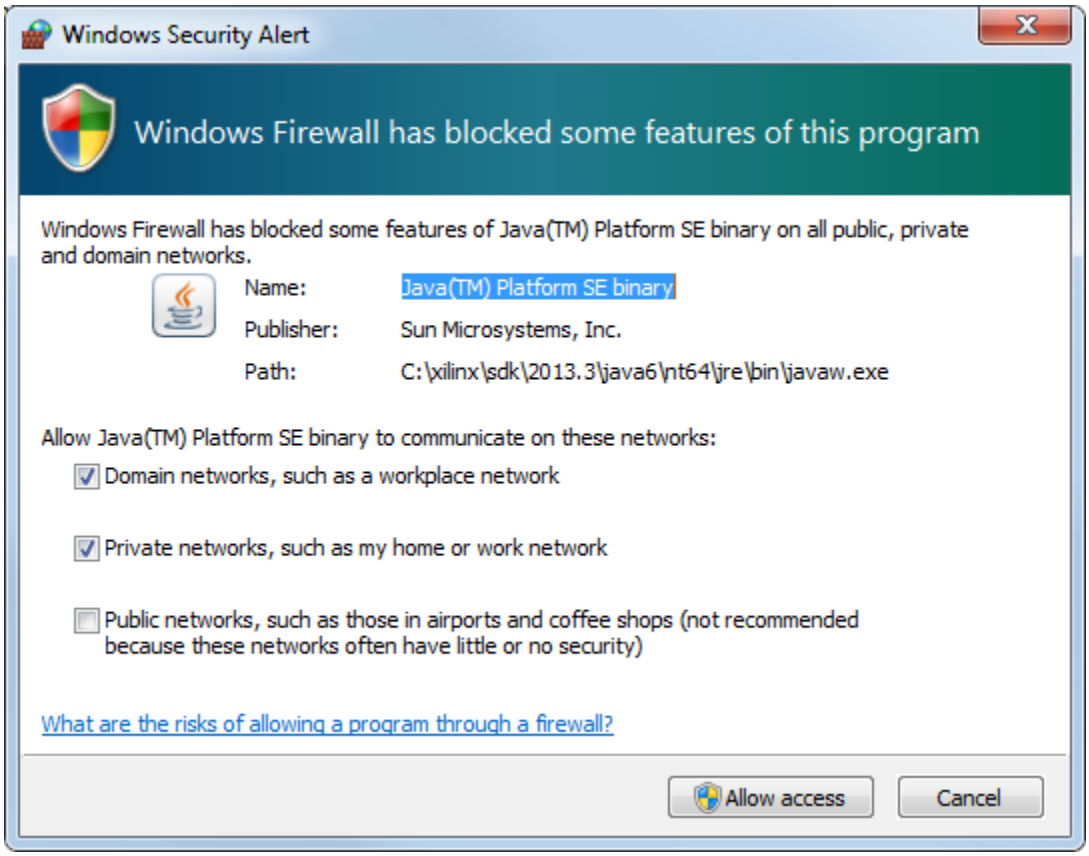

Figure 5 – Allow Java for Xilinx SDK in Windows

Questions:

Answer the following questions:

- *What is the purpose of the SDK Workspace?*

_____

_____

_____

- *When you need to share an SDK Workspace, is it a good idea to simply zip it up and send it over?*

_____

Experiment 2: Examine the Hardware Platform

SDK is now ready for software development, but first we will review the hardware platform peripherals. This is roughly equivalent to reviewing the datasheet for a standard, off-the-shelf processor. The hardware platform specification provides many different important pieces of information, including:

- Peripheral set
- Address map
- Datasheets to peripherals
- System block diagram

Experiment 2 General Instruction:

Examine the hardware platform. Identify the peripherals included in this hardware platform. Review the peripheral address map. Open the Zynq datasheet. Review the external port list. Determine the system clock speed.

Experiment 2 Step-by-Step Instructions:

3. The ***ZynqHW*** hardware platform is now visible under the Project Explorer. If ever not visible, the Project Explorer view can be restored by selecting **Window → Show View → Project Explorer**, or by resetting the entire perspective view with **Window → Reset Perspective**. You should recognize all of these files from the discussion in Lab 1.

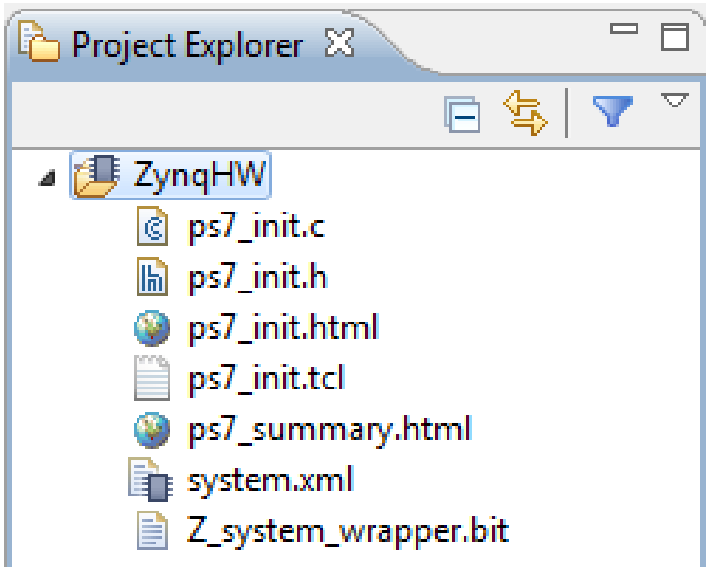

Figure 6 –Hardware Platform Specification in Project Explorer

4. The *Hardware Platform Specification* is already open in the working window since the hardware platform was just imported. To open this at other times, double-click on **system.xml** in Project Explorer under ZynqHW.

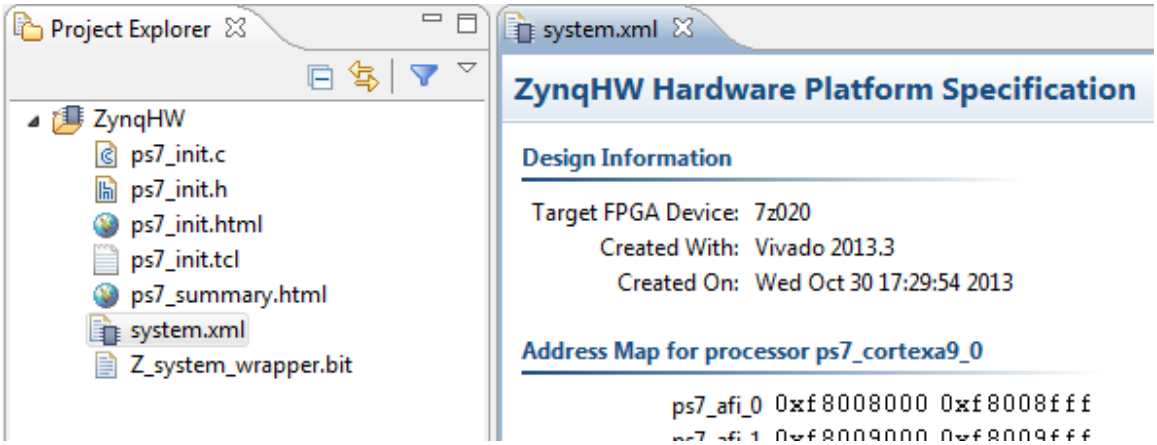

Figure 7 – Hardware Platform Specification

5. The raw XML may also be reviewed. In the lower left, select Source.

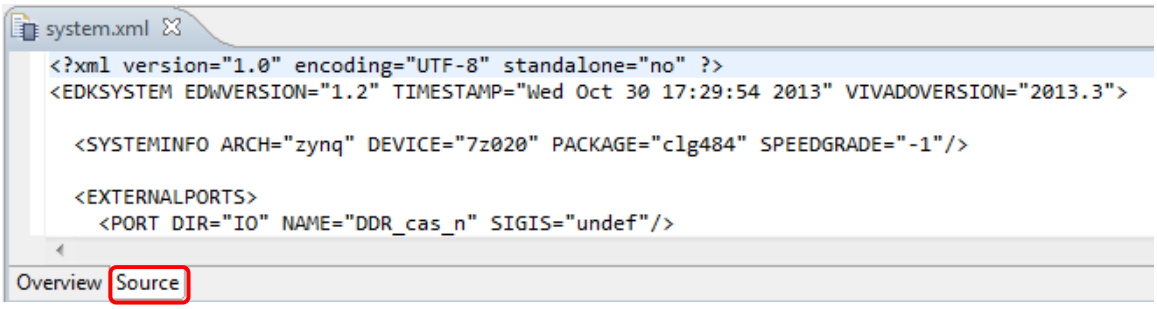

Figure 8 – View the Source XML

6. Compared to reading the raw XML file, the *Overview* report is much more readable. Click on the **Overview** tab to switch back to this view. The information is organized into three sections.

    a. Address Map for processor ps7_cortexa9_0
    b. Address Map for processor ps7_cortexa9_1
    c. IP blocks present in the design

7. Under *Design Information*, note the Target FPGA Device is identified. The version of Vivado that created the hardware platform is also identified.

8. Use these sections to answer the following questions.

Questions:

Answer the following question:

- *What is address range for the DDR3 (ps7_ddr_0)?*

_____

- *How large is the QSPI Flash in the system? Note that peripheral ps7_qspi_0 is the QSPI controller while ps7_qspi_linear_0 is the Flash space.*

_____

- *What is the base address for the PWM Controller with Interrupt (PWM_w_Int_0)?*

_____

- *What is the version of the ARM Processor Core (ps7_cortexa9)?*

_____

9. Browse through system.xml overview in SDK, specifically the ***IP blocks present in the design***. The first column represents the name of the IP in this design. The second column provides the name of the IP. We will focus on the second column and the name of the IP.

10. If you are unfamiliar with any of the peripherals in the design, it would be helpful to access a datasheet. Depending on the type of IP, you may find the IP datasheet in one of three different locations.

    a. Xilinx Processing System IP – the datasheet for these peripherals is found in the Zynq Technical Reference Manual, a copy of which is provided in the `C:\Speedway\ZynqSW\2013_3\Support_Documents` folder.

    b. Xilinx Programmable Logic IP – these are peripherals that are built in the PL fabric. The datasheets for this IP may be found online at www.xilinx.com, or you can make use of the Xilinx Documentation Navigator that is installed with the tools.

    c. Custom Programmable Logic IP – datasheets for custom IP must be obtained from the designer.

11. Use your PDF Reader to open the Zynq Technical Reference Manual (`ug585-Zynq-7000-TRM.pdf`) located in the following folder:

    `C:\Speedway\ZynqSW\2013_3\Support_Documents`

12. Looking at the IP names, choose any of the names that start ps7_*. These are all the Xilinx PS IP. For this document, the ps7_uart is used as an example.

13. In the TRM, search or browse to the UART section. This section explains the hardware IP for the UART Controller. Similarly, you could look in the TRM for descriptions of other PS7_* hardware.

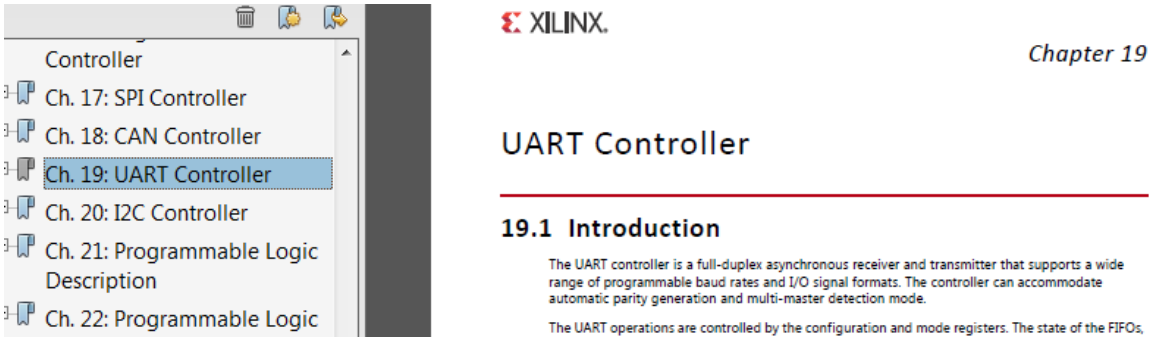

Figure 9 – Documentation on UART Hardware Peripheral

14. The axi_bram_ctrl peripheral in this design is one that is provided by Xilinx. As a piece of IP for the PL, it has its own Product Guide. If you have the full Vivado installed, there is a reference to this Product Guide embedded in the installation. You can find all PL IP listed in the following folder:

    `C:\Xilinx\Vivado\2013.3\data\ip\xilinx`

15. From this directory, browse and open:

    `\axi_bram_ctrl_v3_0\doc\pg078-axi-bram-ctrl.pdf`

You will find this PDF simply points you to www.xilinx.com to get the Product Guide. Unfortunately, the versions are not accurate. Note that this document points you to

http://www.xilinx.com/cgi-bin/docs/ipdoc?c=axi_bram_ctrl;v=v2_00_a;d=pg078-axi-bram-ctrl.pdf

It should be:

http://www.xilinx.com/support/documentation/ip_documentation/axi_bram_ctrl/v3_0/pg078-axi-bram-ctrl.pdf

The axi_bram_ctrl_v3.0_Product_Guide is provided for your reference in the `Support_Documents` folder.

16. You can also search www.xilinx.com directly for “axi_bram_ctrl.” Just be cautious in selecting the correct version. Notice the system.xml lists this IP as v3.0. A search for “axi_bram_ctrl” turns up v1.03.a first, followed by v3.0 second.

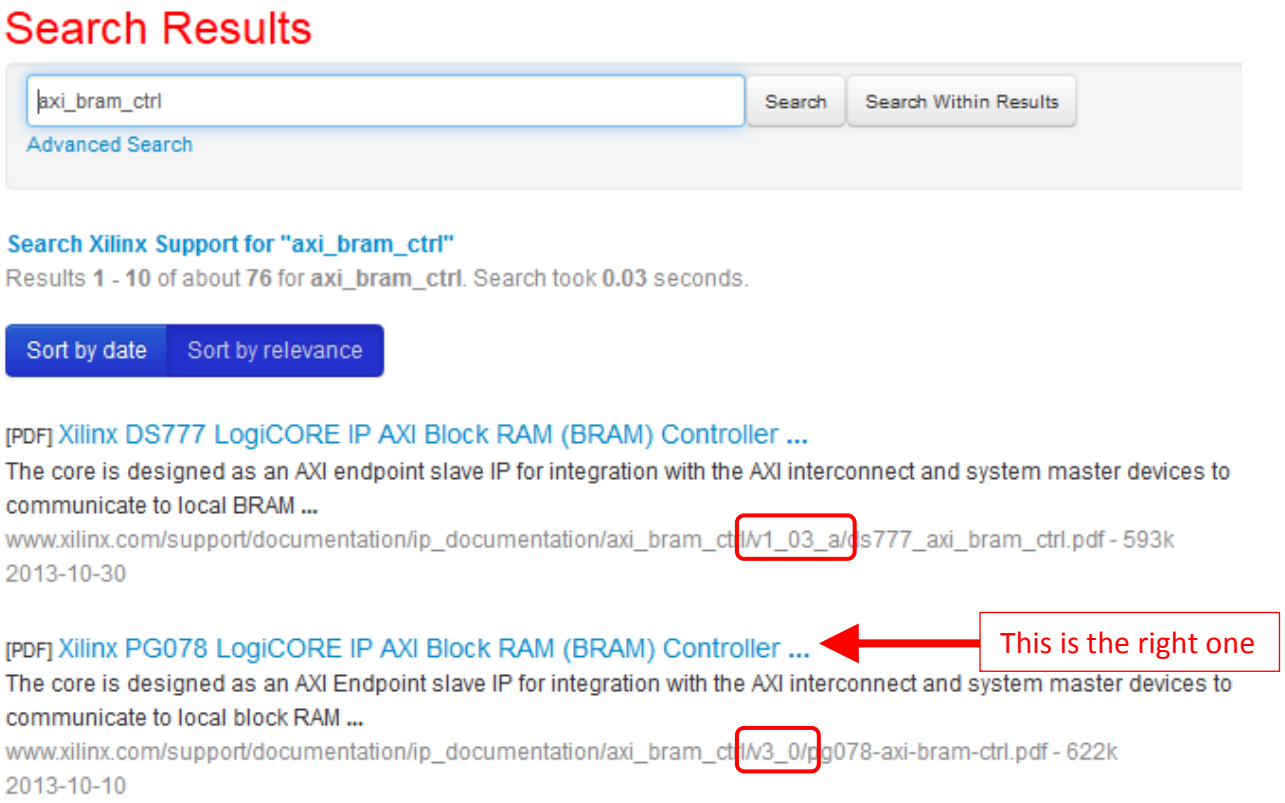

Figure 10 – Search for axi_bram_ctrl

17. The last type of IP is Custom IP. This one is more difficult as documentation for custom IP is a dependent on whoever created the IP. In this design, the PWM_w_Int_0 is a piece of custom IP. You would have to request the documentation from the designer. For this course, the hardware platform and IP is fully documented and explained in the *Developing Zynq Hardware* course.

18. The Vivado tools will also create a graphical block diagram of the hardware design. It may be beneficial to get the hardware engineer to provide a copy of this block diagram.

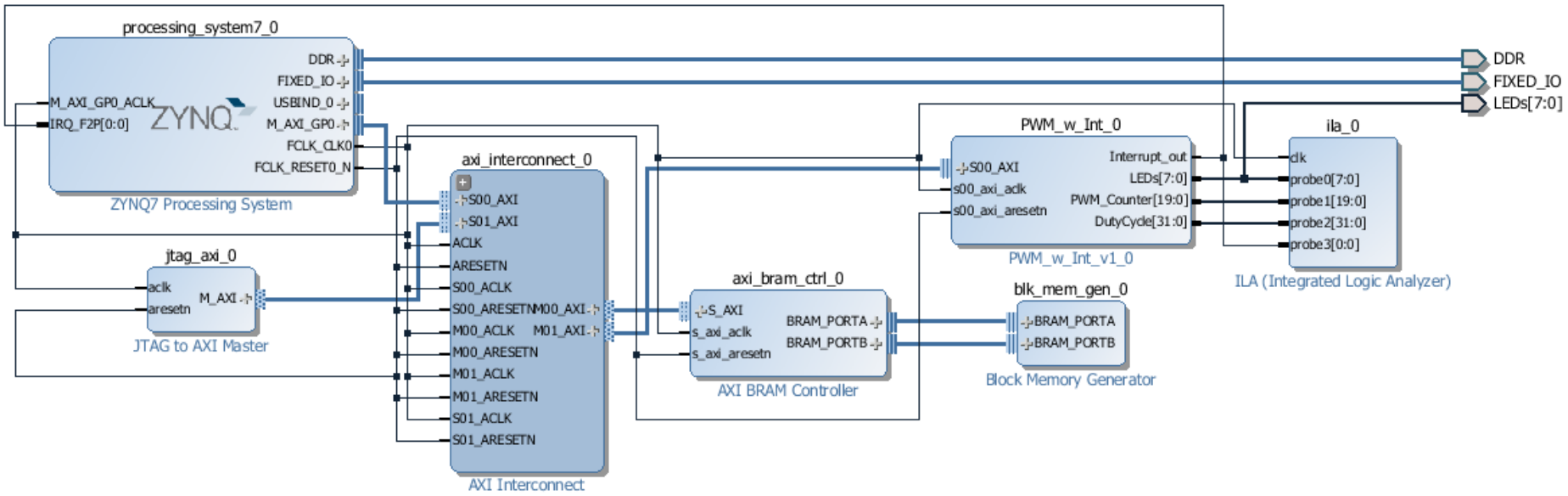

Figure 11 – Hardware System Block Diagram

Question:

Answer the following question:

- *Where will you find the hardware datasheet for the ps7* peripherals?*

_____

Exploring Further

If you have more time and would like to investigate more…

- Review the *Cheat Sheets* available under SDK's **Help** menu.

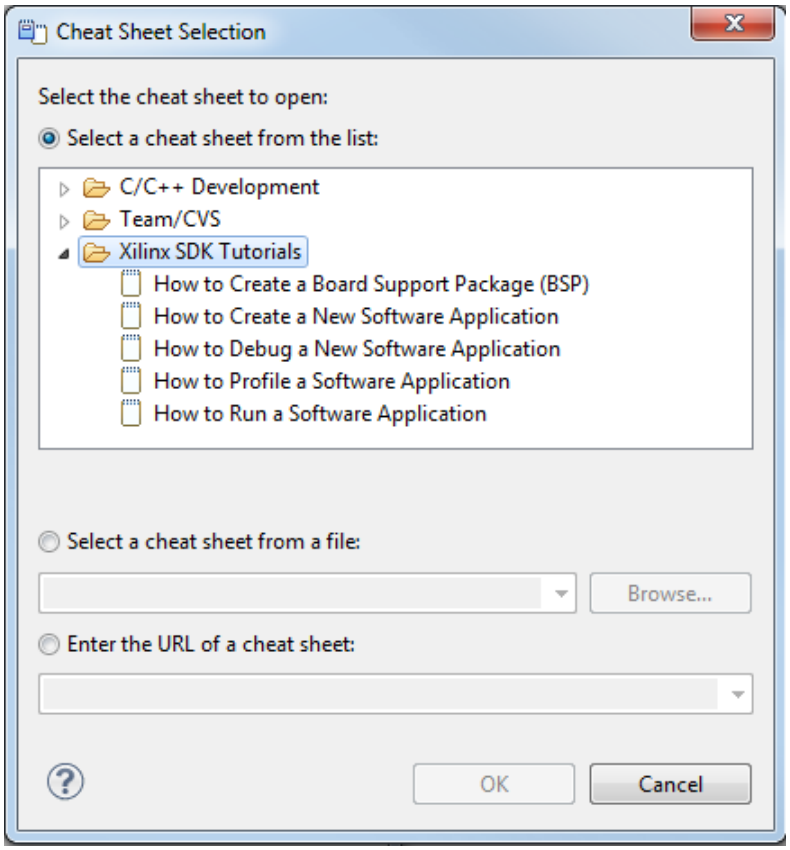

Figure 12 – Cheat Sheets Are Tutorials

This concludes Lab 2.

Revision History

Date	Version	Revision
12 Nov 13	01	Initial release
22 Nov 13	02	Revisions after pilot
01 May 14	03	ZedBoard.org Training Course Release

Answers

Experiment 1

- *What is the purpose of the SDK Workspace?*

The Workspace holds a collection of projects related to your software application development. It will typically contain one hardware platform and one or more BSPs and software applications. The workspace contains your SDK settings, software sources, and logs.

- *When you need to share an SDK Workspace, is it a good idea to simply zip it up and send it over?*

NO! In Lab 8, we'll show you the proper way.

Experiment 2

- *What is address range for the DDR3 (ps7_ddr_0)?*

0x00100000 to 0x1fffffff

- *How large is the QSPI Flash in the system? Note that peripheral ps7_qspi_0 is the QSPI controller while ps7_qspi_linear_0 is the Flash space.*

0xfcffffff – 0xfc000000 + 1 = 0x1000000 = 16 MB = 128 Mb addressable space. ZedBoard actually has a 256 Mb device, but the upper 128 Mb is accessed by flipping a control register, not in linear address space.

- *What is the base address for the PWM Controller with Interrupt (PWM_w_Int_0)?*

0x43c00000

- *What is the version of the ARM Processor Core (ps7_cortexa9)?*

5.2

- *Where will you find the hardware datasheet for the ps7* peripherals?*

The Zynq TRM